Prove con la

Buggy Car

SISTEMA DI MONITORAGGIO GRAFICO DELLA PRESSIONE ATMOSFERICA

- *Chiave d'accesso transponder*
- *Ricezione del GPS con PIC*
- *Orologio per Vinificazione*
- *Display Art*

SOMMARIO

Trova l'articolo di tuo interesse! Questa la legenda:

P Progetti

S Smart Home

D Divulgazione scientifica

Clicca qui per scoprire chi lavora con Fare Elettronica entra anche tu a far parte, contattaci!

L'Editoriale

di Giovanni Di Maria

Ancora novità in casa Fare Elettronica

Questo numero della rivista Fare Elettronica segna ufficialmente il definitivo passaggio al nuovo editore Tecnoimprese. Come sempre, il cambio di editore porta novità e miglioramenti. Un nuovo logo, visibile in copertina e nel colophone, oltre a una grande novità: Fare Elettronica è adesso distribuita gratuitamente, è richiesta una semplice registrazione sul nuovo sito www.farelettronica.com, completamente rinnovato.

Passiamo, adesso, a introdurre gli articoli di questo numero, come sempre molto interessanti, utili e didattici. Apre la rivista Roberto Vallini che, con una descrizione piacevole e divertente, introduce i display intelligenti prodotti dalla 4D Systems. Quindi, l'Ing. Alberto Trasimeni presenta il suo progetto di ricezione dei segnali dai satelliti GPS mediante il microcontrollore PIC18F45K22. Prosegue un interessante progetto di stazione meteo, proposta dall'Ing. Giovanni Carrera, adibita al monitoraggio grafico della pressione atmosferica, con possibilità di previsioni metereologiche. Quindi Moreno Palazzo ci mostra un dispositivo che è più di un giocattolo elettronico: il Buggy Car, prodotto dalla MikroElektronika, completamente programmabile con risvolti sia ludici che didattici. Giuseppe La Rosa illustra una chiave d'accesso utilizzante il trasponder, con cui realizzare molteplici dispositivi per la casa e l'azienda. Chiude la rivista Girolamo D'orio che presenta un utile e innovativo orologio dedicato al processo di vinificazione ma utilizzabile anche in altri ambiti.

Nel rinnovarvi una buona lettura, ricordiamo che Fare Elettronica sarà presente con un suo stand all'importante evento Illuminotronica che si terrà a Padova dal 6 all'8 ottobre 2016, NON MANCATE VI ASPETTIAMO!

Giovanni Di Maria

DISPLAY ART

di Roberto Vallini

In questo articolo tratterò il primo approccio con i display della 4D Systems. Un tutorial, passo dopo passo, per iniziare a “creare”, i progetti più fantasiosi. Questo tutorial, si è reso necessario, vista la scarsissima documentazione in italiano, contrapposta alla grande potenzialità artistica, di una generazione di display, i quali posseggono un loro ambiente di programmazione, ed un linguaggio proprio. Le intenzioni del tutorial sono quelle di dare un aiuto a coloro che si vogliano avventurare nel mondo del “display drawing”, una nuova tendenza che sta spopolando sul Web, ovvero usare dei dispositivi elettronici per creare vere e proprie opere d'arte, alcune delle quali sono addirittura presenti nelle gallerie internazionali.

Figura 1: Questo è display “Diablo” appena collegato al PC

4D Systems “Turning Tecnology into art”

Già il titolo fa presagire qualcosa di nuovo, di creativo, tradotto letteralmente dall'inglese “4D Systems Trasformando la tecnologia in arte”. E' doveroso ricordare i dispositivi “minori”, se così si possono definire, come i “Seven segment” e gli LCD passivi, visto che, oltre ad aver fatto la storia dell'indicazione, hanno aperto la strada a questa nuova tendenza. Da qualche anno la 4D Systems, ha deciso di investire in una serie di display i quali, permettono di liberare la propria fantasia, attraverso una gamma vastissima di prodotti che non solo offrono una visualizzazione ad alta definizione, ma soprattutto delle soluzioni grafiche “custom”, ovvero personalizzate, i cui unici limiti risiedono nella propria fantasia. Se siete dei veri “makers” è ora di fare un salto nel sito della 4D e dare un'occhiata: http://www.4dsystems.com.au/

Il primo approccio

Avviato il software di programmazione ci si trova di fronte ad una schermata che ci chiede quale tipo di display vogliamo utilizzare, personalmente ho scelto il modello “Diablo u220RD 1,3”, visto che ero al mio

Figura 2: Questo è il display sulla basetta artigianale

primo approccio, ho preferito usare per le prove, il più economico, che con 25 euro arriva a casa vostra . Certamente le dimensioni ridottissime, un diametro di circa 3 centimetri, hanno inciso fortemente sulle possibilità progettuali, ma il fine era educativo, e per questo è servito allo scopo.

Le caratteristiche

Il display è gestito dal processore DIABLO16, un integrato altamente performante che gli consente di essere totalmente indipendente da altri dispositivi di programmazione. Se lo si programma in modalità autonoma, occorre utilizzare il suo specifico software, ovvero il 4D Systems Workshop 4 IDE Software. Il Workshop IDE permette di realizzare progetti grafici veramente personalizzati. Il modulo display, attraverso il suo pin out, composto da un header a 30 vie, con pin 2.54mm maschio, può porsi all'altezza di atri device, come Arduino, avendo degli I/O, sia digitali che analogici, i quali permettono ai display della serie di diventare dei veri e propri “Master”. Avendo anche la possibilità di inserire una memoria micro-SD, attraverso un apposito alloggiamento posteriore, non ci sono limiti nelle dimensioni dei file alloggiabili. L'ulteriore pin header di 10 poli, (5 x 2 sovrapposti, con passo 2,54) di cui è dotato, va utilizzato con il cavetto di programmazione originale, (la casa sconsiglia altri adattatori uSB-Serial).

I primi passi

Una volta scaricato il “Workshop IDE 4D Systems”, vi troverete di fronte alla possibilità di 4 diversi tipi di programmazione: “Designer”, “Visi-Genie, “Visi” e “Serial”. La prima difficoltà, è data dal fatto che tutta la documentazione è in inglese. Vi sconsiglio vivamente di fare traduzioni automatiche, del tipo Google Traduttore, non per la scarsa fiducia in questi utili mezzi, ma perché con l'inglese tecnico, potreste trovarvi sorprese incomprensibili. Una per tutte, stavo usando non so quale integrato, un po' di anni fa' decisi di farlo tradurre automaticamente, ebbene, dalla traduzione uscì la seguente frase:” Prima di servire il vassoio, assicurarsi di avere apparecchiato la tavola”. In realtà doveva essere “ Prima di attivare il registro, assicurarsi della congruità della tabella della verità”.

Il posizionamento (Step One)

Innanzitutto la prima operazione da fare, è quella di posizionare il display, in un alloggiamento stabile. Ricordiamoci che stiamo operando con dei dispositivi versatili

Questi link, sono indispensabili per chi si avvicina per la prima volta al mondo 4D Systems, vi troverete sia l'IDE che tutti i manuali necessari per comprendere i fondamenti della creazione dei vostri progetti.

uLCD-220RD Datasheet
http://www.4dsystems.com.au/productpages/DIABLO16/downloads/DIABLO16_serialcmdmanual_R_1_2.pdf
uLCD-220RD 3D PD
http://www.4dsystems.com.au/productpages/uLCD-220RD/downloads/uLCD-220RD_3dpdf_R_1_0.PDF
uLCD-220RD Drawing
http://www.4dsystems.com.au/productpages/uLCD-220RD/downloads/uLCD-220RD_drawing_R_1_0.PDF
uLCD-220RD STEP Mode
http://www.elettroshop.com/uLCD-220RD%20STEP%20Mode
DIABLO16 Processor Product Brief
http://www.4dsystems.com.au/productpages/DIABLO16/downloads/DIABLO16_productbrief_R_1_2.pdf
DIABLO16 Processor Datasheet
http://www.4dsystems.com.au/productpages/DIABLO16/downloads/DIABLO16_datasheet_R_1_4.pdf
DIABLO16 Internal Functions
http://www.4dsystems.com.au/productpages/DIABLO16/downloads/DIABLO16_internalfunctions_R_1_6.pdf
DIABLO16 Serial Command Set Reference Manual
http://www.4dsystems.com.au/productpages/DIABLO16/downloads/DIABLO16_serialcmdmanual_R_1_2.pdf
Application Notes
http://www.4dsystems.com.au/appnotes
Workshop4 IDE
http://www.4dsystems.com.au/product/4D_Workshop_4_IDE

ma altrettanto delicati, con un processore a bordo scheda, che potrebbe essere danneggiato con il solo tocco del dito, attraverso una piccola ma letale scarica di elettricità statica. Ho tagliato uno schermo di plexiglas, con dei distanziali, in maniera da poter lavorare in sicurezza. Nella figura 2, vediamo la base auto-costruita per le prove di programmazione.

Il collegamento (Step Two)

Una volta che il nostro device grafico è ben saldo, occorre dargli un'occhiata dal lato contatti-Pin. Vedremo due GPIO, simili a quelli del Raspberry per intenderci. Il primo ha 30 poli, divisi in due file da 15 e serve per il collegamento Master/Slave oppure host, per chi preferisce il termine. Per connetterlo, occorre un opportuno cavo flat cablato sul connettore GPIO. In questo mio primo progetto, avendo utilizzato il GPIO da 10 poli, per la pura programmazione, mi soffermerò sulla descrizione di questo. Occorre fare molta attenzione, quando si inserisce il cavetto USB-Serial in quanto l'alimentazione andrebbe a finire sul Reset. Il cavo ha un driver proprietario, che si deve scaricare, sempre dal sito della 4D indicato precedentemente. Di seguito la disposizione dei pin dell'uLCD220RD:

1	VDD -5V
2	Not Conn.
3	Tx
4	Not Conn.
5	Rx
6	Not Conn.
7	GND
8	Not Conn.
9	Reset
10	Not Conn.

L'IDE della 4D Systems Workshop

Giunti a questo punto, se avete seguito le indicazioni per il download dei file necessari, vi troverete sullo schermo del vostro pc, l'icona del workshop, ovvero dell'ambiente di programmazione. Cliccate sull'icona e si aprirà la finestra che vedete in Figura 5, andate sul menù a tendina della barra destra e scegliete “NEW”, la quale vi reindirizzerà verso un altro menù a scorrimento dove compaiono i vari tipi di display. Scegliete il vostro e si parte per l'avventura!

Visi-Genie

Questa è sicuramente la modalità di programmazione più intuitiva per l'utente, essa richiede il cavo di trasduzione USB-Serial, ma ha il vantaggio di non usare codici specifici, se non in fase avanzata. Dopo aperto il workshop 4D ci troviamo di fronte ad una toolbar intuitiva, con una serie di oggetti grafici che vanno dai primitivi poligoni:

(cerchio, triangolo, rettangolo, linea ed ellisse), passando per le “labels” ovvero etichette e segnalibri, compilabili con un testo a vostro piacimento, e arrivando ai “gauges”, (manometri, indicatori di velocità, v-meter, ecc.). Non mancano indicatori di ogni tipo e addirittura animazioni e fotografie, le quali possono essere personalizzate, creando un ricordo multimediale, importando file a vostro piacimento, ma questo lo vedremo nella pratica dei tre progetti realizzati.

Il mio primo progetto “Jack the smoker”

Per il mio primo progetto ho utilizzato la modalità Visi-Genie, che tra le quattro disponibili è sicuramente la più intuitiva.
Programmare con Visi-Genie (Step Three) a “Man-Jack the smoker ”. Come anticipato nel capitolo “I primi passi”, questa modalità è la più intuitiva, presenterò qui tre progetti grafici, in maniera da rendere, più educativo il primo approccio. Dunque, adesso che avete scelto il display vi troverete di fronte ad una nuova schermata con un nuovo menù, per intenderci, una toolbar di lavoro. In alto dalla quale potete selezionare il tipo di oggetto da posizionare nel vostro display. Compare vuota a destra, con una serie di indicazioni di posizione. Le cartelle in alto, contengono nove possibilità di scelta ,“button”, “gauges”, “labels” e altri oggetti in sequenza. Scegliete la cartella “primitive”, cliccate e selezionate: “triangle”, spostatela sul display e sistematela a vostro piacimento. Nel progetto 1 ho trascinato, dopo averle cliccate, quattro sagome, nell'ordine: due cerchi, un triangolo e un ellisse, per realizzare una simpatica figura che ho battezzato “Jack the smoker”, visibile in Figura 4.

Figura 3: L'IDE del 4D Workshop

Il mio secondo progetto “Ufo Invasion”

Sempre nell'ottica della creazione, ma introducendo alcuni nuovi passaggi obbligati, che vi troverete ad affrontare , nella modalità Visi-Genie, ho immaginato un' ”invasione aliena”, se vi sembra follia è invece un esempio di come, la vostra fantasia sia messa alla prova con i display della 4D. Seguendo le medesime procedure descritte nel primo progetto “Jack the smoker” ho realizzato, sempre con alcune figure a disposizione un “UFO” multicolore.

Introduzione alla scelta cromatica

Se avete inserito qualunque oggetto, esso avrà una colorazione propria. Sorpresa delle sorprese, il workshop, contiene una parte interamente dedicata alla cromatica. Facciamo un esempio pratico, il cerchio è verde, se lo posizionate, vedrete comparire un menù chiamato “color”, cliccando la tendina, avrete accesso a tre gamme di colori: Windows colors, Theme Colors e 4DGL Colors. Ogni gamma contiene circa trenta variazioni. Ma non basta, se siete degli artisti esigenti, come è giusto che sia, in “color Picker” potrete scorrere una finestra con tutti i colori visibili. Questo è veramente il massimo. Nel menù di destra, oltre ad avere le coordinate degli oggetti, sull'asse X e Y, avrete anche le dimensioni, espresse in

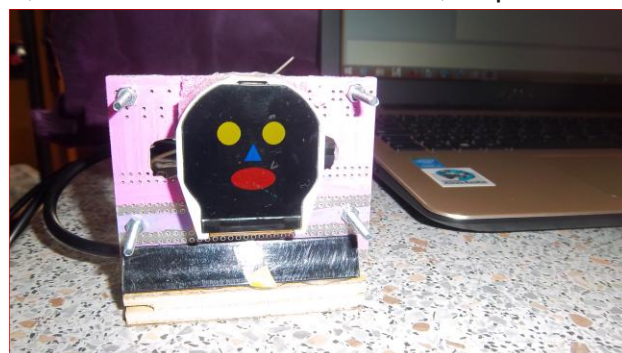



Figura 4: My “Jack the smoker”

millimetri che, se modificate, fanno cambiare forma e posizione. Viceversa, le dimensioni si possono cambiare attraverso una procedura simile a quella usata dal comunissimo programma "Paint", quindi trascinando i lati, senza preoccuparsi, delle dimensioni indicate sugli assi delle ordinate e delle ascisse.

Il mio Terzo progetto "Barbara", ovvero l'apoteosi della creazione

Avendo conservato gelosamente, le fotografie dei momenti più belli della mia vita, in questi giorni, dalla rigorosa scatola dei ricordi, è saltata fuori una fotografia di una bellissima ragazza, la quale negli anni, è divenuta mia moglie. "Perché non metterla sul display"? Ho pensato. Impossibile? Niente diventa impossibile con la 4D! Nella toolbar degli oggetti inseribili, la prima folder è dedicata a oggetti multimediali, "ergo" - mi sono detto - "Fisserò questo splendido ricordo, di una gita scolastica sul display". Ecco come ripetere ciò che ho fatto:

1) Percorrere la medesima routine degli altri due progetti;
2) Giunti alla cartella "Home", selezionare Systems/Media e di seguito "Image";
3) Si aprirà una scelta di cartelle presenti nel vostro pc, selezionate quella dove conservate le immagini più preziose;
4) Cliccate su quella che gradite;

Figura 5: My "UFO Invasion"

5) Trascinate l'immagine sul "portrait" (Lo schermo del display può essere ruotato a piacimento);
6) A questo punto, la fotografia sarà traslata sullo schermo, poi cliccate la radice del file nella nella tendina di destra;
7) Si aprirà una finestra con l'intestazione "Image converter", nella quale potrete apportare modifiche cromatiche e dimensionali, a vostra scelta, dopodiché cliccate sul pulsante "OK" e la vostra amata fotografia sarà sul display.

Fino a qua mi son sbizzarrito nell'arte, con delle creazioni adatte alle dimensioni del piccolo "Diablo", le quali, dimostrano come questo display, possa uscire, dalla stretta logica della segnalazione, e si possa immergere in progetti fantasiosi.

Inserimento di oggetti non primitivi

Gli indicatori possono essere utili per temperatura, luce, posizione e altri tipi di indicazione. Qui la cosa si complica leggermente, infatti, come già detto l'IDE offre una svariata serie di possibilità. Gli indicatori, a differenza degli oggetti "Prmitives", sono più "pesanti", nel senso che richiedono dei banchi di memoria più ampi, quindi i 6 banchi di memoria flash da 32.750 byte, per il codice utente e dati ed i 32Kb di SRAM puri, per l'utente non bastano più. Occorre munirsi di una microSD card, con almeno 2 Giga di spazio, e di un lettore di micro SD esterno al PC, affinché tutta la routine diventi più semplice possibile. Si parte sempre dalla maschera iniziale "NEW", si sceglie, nel menù a tendina il tipo di display, quindi si apre la modalità Visi-Genie. Nella schermata "Home" si va a selezionare un indicatore, ad esempio, nel menù "Gauges", un indicatore circolare di temperatura. Fatto questo passo, occorre, nel menù"project", deselezionare Run-Flash, ossia la memoria a bordo del display e selezionare microSD. Questo passaggio è di fondamentale importanza, viceversa vi troverete dei continui "bug" del sistema.

Questa parte richiede concentrazione e non basta, visto che, pur facendo le opportune operazioni, potrebbe non andare a buon fine. Questo problema non è vostro. Una comunicazione ufficiale della 4D, dopo ripetute segnalazioni, informa che non tutte le microSD sono accettate per la programmazione, attraverso il forum è stato pubblicato il seguente elenco di card non accettate. Quante ore di lavoro perse, prima di capire che non ero io a sbagliare… Una volta risolto l'arcano, posizionate l'oggetto, usando l'opzione download del medesimo, qui comparirà un minaccioso messaggio che dice di mettere il progetto in una periferica diversa. Nessun problema, se ci siamo premuniti del lettore esterno, si carica il progetto sul driver indicato, che potrà essere E, oppure F, a seconda delle periferiche di memoria esterne, inserite nella presa USB del pc, e azionare “Copy-Download”. Fatto ciò si dovrà estrarre la card dal lettore ed inserirla nell'apposito alloggiamento del display. Quindi, come conseguenza, avremo il nostro indicatore di temperatura piazzato e visibile. La tratteggiatura verde che compare a ogni piazzamento, è molto utile, in quanto indica che si sta sbordando dai confini del display. Muovendo il puntatore, si possono modificare le coordinate fino alla scomparsa della tratteggiatura e alla conclusione della routine. Buon divertimento.

Conclusioni

Questo mio articolo, è il frutto di una meticoloso studio sui display 4D ,Systems sul workshop e sulle modalità di connessione e programmazione. Lo scopo di quanto scritto è quello dare, a chi si affaccia per la prima volta in questo “mondo”,uno strumento di base per orientarsi. Infatti per poter esplicare tutte le modalità e le possibilità, non basterebbe un tomo di 300 pagine. Io ne ho scaricate e tradotte 157. Considerato che sul web, in lingua italiana c'è pochissimo, i dati a disposizione, riguardano essenzialmente le caratteristiche tecniche. Ritengo di aver impostato un tutorial, comprensibile ma non troppo pesante.

Desumo che tutto ciò, frutto di tantissime ore di traduzione e studio abbia una valenza funzionale. Chi, come me, vive nel mondo dell'elettronica, forse, guarda con un pizzico di nostalgia, scomparire i circuiti “seven segment”, con la loro imponente circuiteria, e i display LCD a righe e colonne, ma è inevitabile affacciarsi ai progressi della tecnologia, puntando ad essere sempre i primi.

Figura 6: My “Barbara project”

Alcune microSD non sono accettate dal display:

External label, Manuf ID (Hex), Manuf ID (Text), OEM/Appl ID, PRoduct Name, Rev, Serial, Manuf Date
Kingston 2gb, 0x1B, Samsung, SM, 00000, 1.0, 0EB71E78, 12/2011
noname 1gb, 0x1B, Samsung, SM, 00000, 1.0, B1169098, 8/2008
Transcend 8gb, 0x74, Unknown, JE, USD, 1.0, 4E840E92, 10/2012
Transcend 2gb, 0x1B, Samsung, SM, 00000, 1.0 3762D0E0, 12/2012
KingMax 64mb, 0x13, Kingmax, KG, SD064, 1.0, 00014572, 9/2005
noname 2gb, 0x12, Unknown, Vx ASTC, 3.4, 00000000, 2/2002
Kingston 2gb, 0x02, Toshiba, TM, SD02G, 6.0, AB17ACA2, 6/2010
Kingston 4gb, 0xAA, Unknown, (null), (null), 0, EEC29C1C, 1/2011

RICEZIONE DEL GPS MEDIANTE PIC18F45K22

Ing. Alberto Trasimeni

Il Global Positioning System generalmente chiamato GPS è un sistema che permette di individuare la posizione di un ricevitore su tutta la superficie terrestre, basato su un sistema satellitare a copertura globale, ed è gestito dal Dipartimento della difesa statunitense. All'inizio degli anni 90 gli Stati Uniti introdussero il servizio di nome SPS (Standard Positionig System) con specifiche diverse da quelle usate per scopi militari, anche chiamato PPS (Precision Positionig System). La differenza tra i due era dovuta alla Selective Availability, che introduceva intenzionalmente degli errori nei segnali satellitari ad uso civile, al fine di ridurre l'accuratezza della posizione rilevata con un errore dai 100 ai 150 m. Tale degradazione dei segnali satellitari è stata eliminata nel maggio 2000 da Bill Clinton, mettendo così a disposizione anche per l'uso civile la precisione attuale di circa dai 10 ai 20 m. Attualmente nel firmamento satellitare sono presenti 32 satelliti attivi ed altri supplementari che migliorano la precisione del sistema di misurazione della posizione.

Figura 1: Il sistema GPS.

Il Global Positioning System generalmente chiamato GPS è un sistema che permette di individuare la posizione di un ricevitore su tutta la superficie terrestre, basato su un sistema satellitare a copertura globale, ed è gestito dal Dipartimento della difesa statunitense. All'inizio degli anni 90 gli Stati Uniti introdussero il servizio di nome SPS (Standard Positionig System) con specifiche diverse da quelle usate per scopi militari, anche chiamato PPS(Precision Positionig System). La differenza tra i due era dovuta alla Selective Availability, che introduceva intenzionalmente degli errori nei segnali satellitari ad uso civile, al fine di ridurre l'accuratezza della posizione rilevata con un errore dai 100 ai 150 m. Tale degra-

dazione dei segnali satellitari è stata eliminata nel maggio 2000 da Bill Clinton, mettendo così a disposizione anche per l'uso civile la precisione attuale di circa dai 10 ai 20 m. Attualmente nel firmamento satellitare sono presenti 32 satelliti attivi ed altri supplementari che migliorano la precisione del sistema di misurazione della posizione. GPS è strutturato nelle seguenti componenti, che partecipano assieme alla gestione del sistema di rilevazione della posizione, ed esattamente:

1) Un insieme minimo di 24 satelliti operativi, divisi in gruppi di quattro e disposti su sei piani orbitali,distanti 60° fra di loro ed inclinati di 55° rispetto al piano equatoriale;
2) Ognuno di questi satelliti deve compiere due rivoluzioni terrestri al giorno;
3) Una rete di stazioni per il tracciamento (tracking station);
4) Un centro di calcolo (computing station);
5) Due stazioni di soccorrimento (injection station);
6) Un ricevitore GPS.

Il principio di funzionamento si basa sul metodo del posizionamento sferico, che consiste nel misurare il tempo impiegato da un segnale radio per percorrere la distanza satellite-ricevitore. Conoscendo il tempo impiegato dal segnale per giungere al ricevitore e l'esatta posizione di almeno tre satelliti per individuare una posizione bidimensionale, e quattro per avere una posizione tridimensionale, è possibile determinare la posizione nello spazio del ricevitore stesso, come mostrato in figura 1. Questo metodo è chiamato anche trilaterzione ed è del tutto simile alla triangolazione, e da questa si differenzia perché si può fare a meno di informazioni riguardanti gli angoli. La precisione può essere considerevolmente aumentata grazie all'uso di sistemi come il WAAS (tecnologica statunitense) o l'EGNOS (tecnologia europea) entrambi

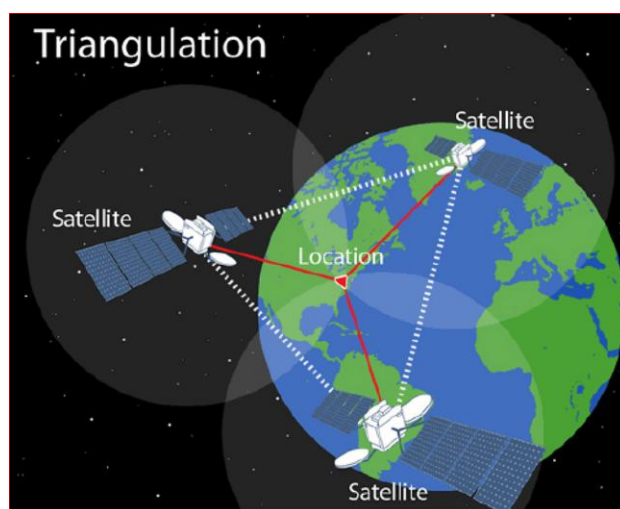



Figura 2: La triangolazione.

perfettamente compatibili tra di loro. Queste due tecnologie consistono in uno o due satelliti geostazionari che inviano segnali di correzione. È presente anche la modalità DGPS (Differential-GPS) che utilizza un collegamento radio da una stazione terrestre per ricevere dati al fine di ridurre l'errore di rilevazione della posizione di circa due metri. Esistono in commercio diverse tipologie di ricevitori GPS esterni interfacciabili con diversi protocolli,che utilizzano porte USB, I2C,eccetera. La figura 2 evidenzia la triangolazione. Le applicazioni del GPS sono diverse, una rilevante è quella che in gergo tecnico viene chiamata WSN (Wireless Sensor Network) che è un sistema complesso, che nasce dalla interazione tra diversi soggetti elementari chiamati nodi sensori. Si tratta nella sostanza di veri e propri sistemi embedded, dispositivi elettronici in grado di svolgere autonomamente un insieme di azioni più o meno complesse, di interagire con l'ambiente esterno e di cooperare tra di loro per mezzo di opportune interfacce di comunicazione. Tutto ciò e dovuto alla necessità di poter disporre di reti senza infrastruttura e senza controllo centralizzato, perciò sono semplici ed economiche da realizzare.La WSN introduce nuovi protocolli di routing che si basano sulle informazioni geografiche dei nodi,come una valida alternativa agli attuali protocolli per reti wireless,da qui la necessità di conoscere la posizione geogra-

fica dei nodi e quindi l'uso del GPS. La comunicazione tra i vari nodi avviene utilizzando onde radio. Ricercatori dell'università di Padova, in questi ultimi anni,hanno concentrato i loro sforzi nello sviluppo di queste nuove tecnologie (laboratorio SIGNET). Per realizzare il collegamento GPS utilizzeremo come materiale hardware, il sistema di sviluppo EasyPIC7 con il PIC18F45K22, ed il ricevitore GPS_click della MikroeleKtronica. Il ricevitore monta un modulo GPS LEA-6S della Ublox, di ultima generazione con alto grado di performance in svariate condizioni ambientali e dotato di un basso consumo. Il protocollo scelto per la comunicazione con il ricevitore è rappresentato dalla seriale asincrona che è raggiungibile utilizzando il miKroBUS presente sulla scheda EasyPIC7, rappresentato da un doppio connettore a pettine con passo 2,54mm, come mostra la figura 3. L'accensione e lo spegnimento del modulo avvengono accendendo e spegnendo la scheda EasyPIC7, in quanto il modulo ricevitore non è dotato di interruttore di accensione e spegnimento. La massima tensione di lavoro è di 3,6V per cui la EasyPIC7 è dotata di un jumper che permette di passare dalla classica alimentazione di 5V a quella di 3,3V, questa operazione deve essere fatta prima di inserire il modulo sulla scheda onde evitare il suo danneggiamento. Il modulo per ricevere correttamente i segnali dai satelliti deve essere collegato all'antenna

Figura 3: Il ricevitore GPS_click.

utilizzando l'apposito connettore visibile in figura 3. Questa è preferibile posizionarla all'esterno degli edifici, per avere dati validi, altrimenti si potrebbero creare dei ritardi sui segnali inviati dai satelliti, per effetto dei diversi ostacoli incontrati nel percorso, dando origine a ripetute riflessioni ed attenuazioni sui segnali stessi. Ogni volta che il modulo GPS viene acceso, carica le impostazioni di default che usa per inviare i dati che ha elaborato al microcontrollore a cui è collegato tramite il miKroBUS. In particolare, la seriale asincrona viene configurata per trasmettere e/o ricevere dati alla velocità di 9600 baud con un bit di start,uno di stop e nessuna di parità. Ogni secondo e senza pause vengono inviate sulla linea "Tx" del modulo, in un formato speciale e standardizzato dal National Marine Electronics Association (NMEA) le seguenti stringhe di dati:

1) GGA (dati fissi per il Global Position);
2) GLL (posizione geografica- Latitudine/Longitudine);
3) GSA (degradazione dell'accuratezza e di numero di satelliti attivi nel sistema di navigazione satellitare globale;
4) GSV (satelliti visti dal ricevitore);
5) MSS (rapporto segnale rumore e la frequenza e velocità di 1 bit ricevuto via radio;
6) RMC (ora e data, posizione e velocità dei dati);
7) VTG (direzione, velocità orizzontale);
8) ZDA (UTC time, giorno, mese, anno, ora locale e minuti locali);
9) Ok di fine messaggio.

La figura 4 mostra i dati trasmessi dalla seriale del GPS al PIC18F45K22 utilizzando l'oscilloscopio CircuitGear della Syscomp.

Per ricevere e visualizzare su LCD 20x4 le informazioni provenienti dal modulo

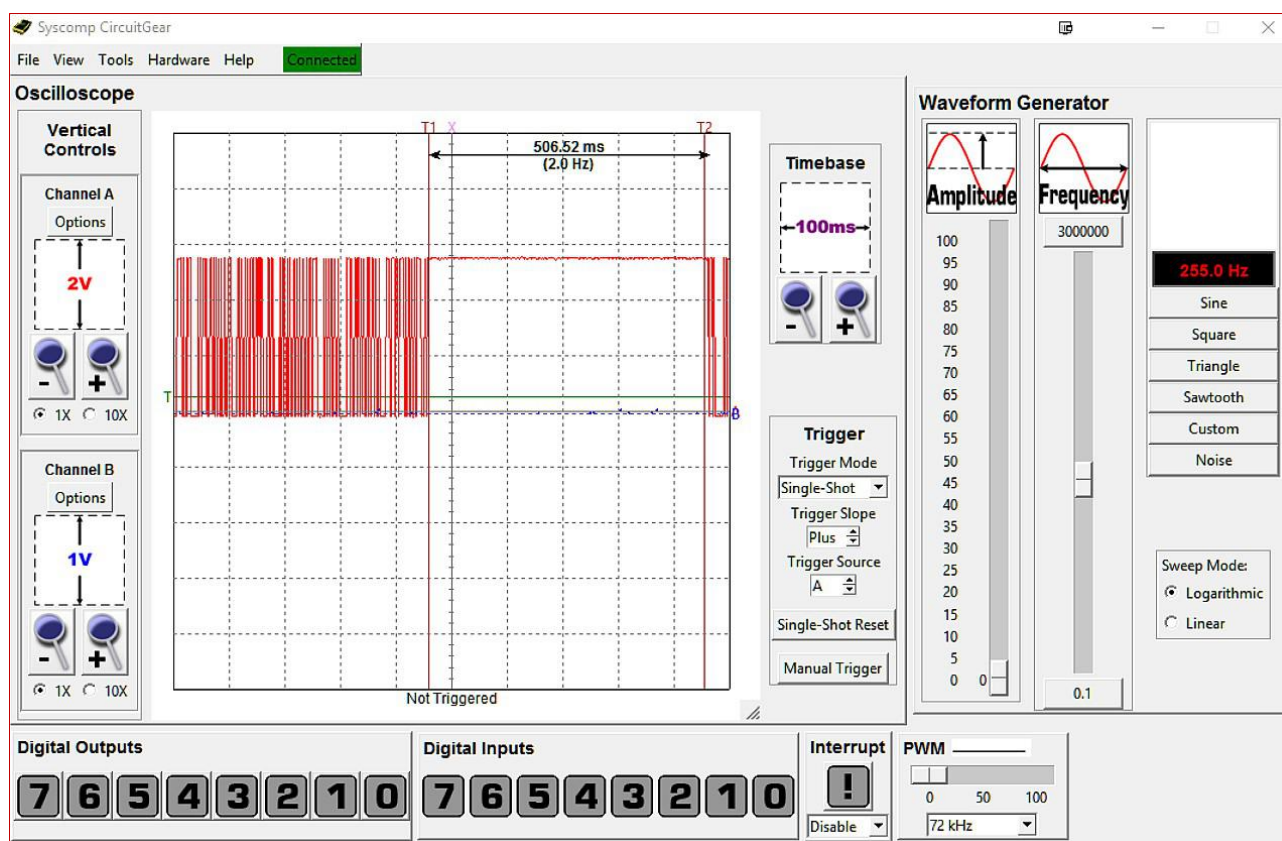



Figura 4: I dati trasmessi dalla seriale del GPS al PIC18F45K22 registrati con l'oscilloscopio CircuitGear della Syscomp.

GPS, è necessario, dopo aver riconosciuto la stringa d'interesse, effettuarne il "parsing" cioè individuare la posizione dei campi che vogliamo utilizzare all'interno di essa. Quindi, la prima cosa da fare è quella di analizzare la sua struttura. Come detto in precedenza, i dati sono trasmessi nello standard NMEA,che utilizza caratteri ASCII ad 8 bit. Ogni stringa in ricezione, possiede la seguente forma:

"Tipo": $GPTipo, campo_1,campo_2, campo_3,..., campo_n*CS<CR><LF>.

Il significato dei singoli caratteri o il set di caratteri è esplicato qui sotto:

Campo **Descrizione**

$ Inizio del set di dati
GP Informazioni dal GPS
Tipo Identificatore del set di dati
campo_1 Informazioni contenute nel set
I Separatore fra le informazioni
* Separatore per il checksum
CS Checksum
CR+LF Fine del set di dati

Il numero massimo di caratteri, presenti nella generica stringa, non deve essere maggiore di 80, e nel conteggio dei caratteri, non devono essere considerati il carattere di inizio stringa "$" e quelli di terminazione "<CR>" e "<LF>", cioè carriage return e line feed. L'insieme di tutte le stringhe inviate dal modulo GPS è chiamato "sentenza".
Il firmware di ricezione del PIC18F45K22 lavora su due stringhe ed esattamente:

1) "$GPZDA" (per determinare orario e datario);
2) "$GPGGA" (per determinare le coordinate geografiche Latitudine, Longitudine e numero dei satelliti in vista).

Per ragioni di spazio, illustrerò solo la struttura della stringa "GGA" che contiene le informazioni geografiche, come da esempio e tabella:

Campo	Descrizione
GGA	Identificativo protocollo GGA
195044.00	Ora UTC (hhmmss.ss)
4508.43100	Latitudine (ddmm.mmmmm)
N	N = nord; S = sud
00737.21937	Longitudine (dddmm.mmmmm)
E	E = est; W = ovest
1	Qualita del Fix: 0 = Invalido 1 = GPS fix 2 = DGPS fix 3 = Fix GPS PPS 4 = RTK (Real Time Kinematic) intera 5 = RTK float 6 = Navigazione Stimata (dead reckoning) 7 = Input Manuale 8 = Simulazione
06	Satelliti in vista
1.5	HDOP
278.4	Altitudine rispetto al livello del mare
M	Unita di misura metri
47.2	Separazione geoidale
M	Unita di misura metri
vuoto	Tempo dall'ultimo aggiornamento DGPS
vuoto	Id della stazione DGPS
*5F	Checksum
<CR><LF>	Caratteri di terminazione

Tabella 1: Struttura della stringa "GGA"

$GPGGA,195044.00,4508.43100,N,00737.21937,E,1,06,1.69,278.4,M,47.2,M,,*5F <CR><LF>

Voglio precisare che ho usato anche la stringa "ZDA",in quanto la "GPA" pur contenendo anch'essa l'ora, non conteneva il datario che volevo visualizzare su LCD. Per determinare la posizione delle informazioni all'interno della stringa, sarà sufficiente usare il semplice stratagemma mostrato in figura 4a. Inoltre è possibile inviare opportuni comandi al modulo GPS, mediante il protocollo NMEA, al fine di configurarlo, come ad esempio disabilitare l'invio di alcune stringhe o cambiare il baud rate della seriale. La struttura del comando (tabella 2) è la seguente:

Campo	Descrizione
$PUBX	UBX protocol header
N	Tipo di comando
par_1,..., par_N	Parametri del comando
CS	Checksum
<CR><LF>	Caratteri di terminazione

Tabella 2: Struttura del comando.

"$PUBX,N,par_1,...,par_N,*CS<CR><LF>".

Riporto per comodità alcuni comandi di configurazione del modulo GPS in tabella 3. Il CRC è compreso nella stringa di commando è può essere facilmente calcolato, come l'XOR di tutti i caratteri, a partire dal primo, dopo il carattere di inizio stringa ($), e fino all'ultimo prima del terminatore (*). Oppure utilizzando il seguente link di internet:

http://www.hhhh.org/wiml/proj/nmeaxor.html

Per effettuare la simulazione del progetto mediante Proteus rel.7.10 mi sono servito di un ottimo simulatore del GPS prodotto dalla Sailsoft, di nome Nemastudio,che prevede diversi settaggi, come ad esempio il WAAS per aumentare la precisione nella determinazione della posizione del ricevitore. La sentenza GPS inviata su seriale asicrona, da parte del simulatore, è basata su stringhe NMEA 0183 perfettamente compatibili con il ricevitore LEA-6s della Ublox utilizzato dal GPS-Click della MikroEleKtronica e da me impiegato. Inoltre è necessario installare il programma "Virtual serial port driver" di Eltima software. In figura 5 e 6 riporto lo schema

Significato Del Comando	Struttura Del Comando
Cambio velocità da 9600 a 115.200	$PUBX,40,1,0007,0003,115200,0*18<CR><LF>
Disabilitare stringa "GLL"	$PUBX,40,GLL,0,0,0,0,0,0*5C<CR><LF>

Tabella 3: Alcuni comandi di configurazione del modulo GPS.

```
0123456789012345678901234567890123456789012345678901234567890123456789
$GPGGA,121429.50,3001.84101,N,02500.78146,E,1,12,0.0,0,M,50.0,M,0,0*79
```

Figura 4a: Un semplice stratagemma per determinare la posizione delle informazioni all'interno della stringa.

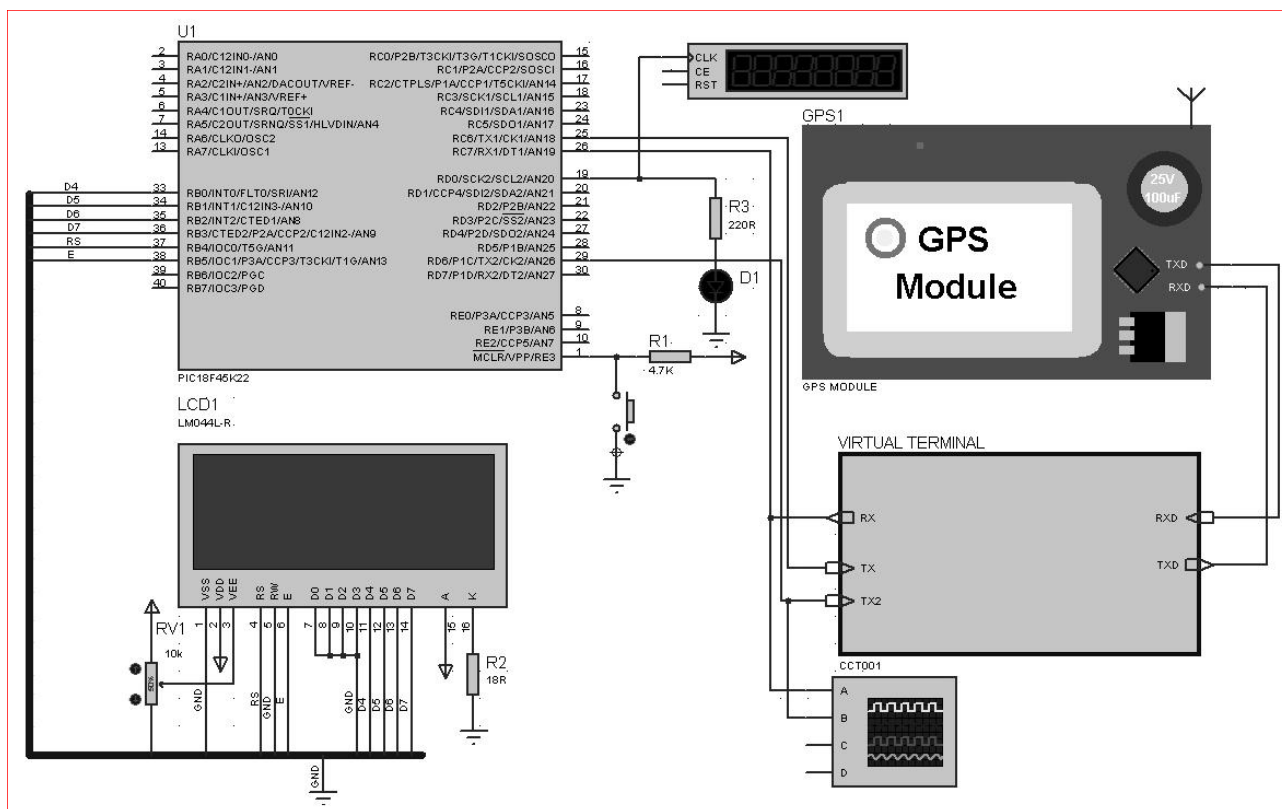



Figura 5: Schema elettrico.

di principio realizzato con Proteus e la simulazione ottenuta, in cui si evidenzia le stringhe inviate dal simulatore al PIC18F45K22.
In figura n°6 si nota, sul display lcd, oltre l'UTC (time) anche l'UTC (time+1.00) del nostro fuso orario, il datario il numero dei satelliti in vista e le coordinate latitudine e longitudine, in gradi, primi e minuti primi, dalla lunghezza numerica di quest'ultimi dipende la precisione con cui viene individuata la posizione del ricevitore(target).
Illustrerò ora il software di gestione del progetto e in modo particolare:

1. La funzione di ricezione della sentenza del GPS;
2. Il parsing della stringa "GGA" per determinare latitudine e longitudine;
3. Il parsing della stringa "ZDA" e come determinare orario e datario.

La funzione di ricezione dell'intera sentenza del GPS utilizza l'interrupt della seriale rs232, in effetti il microcontrollore possiede due seriali entrambe da me utilizzate, una per la ricezione dei dati da parte del ricevitore GPS e l'altra per l'invio delle diverse stringhe NMEA al PC. A tale scopo ho usato il programma RealTerm (per il PC) ottimo emulatore di terminale per catturare i dati inviati dal PIC, come mostra la simulazione ottenuta con Proteus (figura 6).
Nel listato 1 è presente la gestione dell'interrupt per la seriale (1) e l'invio dei dati alla seriale (2) monitorati dal PC. Una volta ricevuti i 768 byte che costituiscono l'intera sentenza del GPS, viene settato il flag di fine ricezione, ready=1, che consentirà al programma principale di effettuare il parsing delle stringhe ZDA e GGA, la prima contenente le informazioni relative all'orario e al datario, la seconda le coordinate geografiche di posizione del ricevitore.
Facendo uso del puntatore "string" è possibile stabilire se nel buffer di ricezione è presente la stringa NMEA GGA, ed una volta individuata, mediante la funzione memcpy viene trasferito nei campi, latitudine e longitudine i relativi valori, che verranno successivamente calcolati, in relazione alla posizione del ricevitore sulla superficie terrestre, come mostra il listato 2. Sempre utilizzando il puntatore

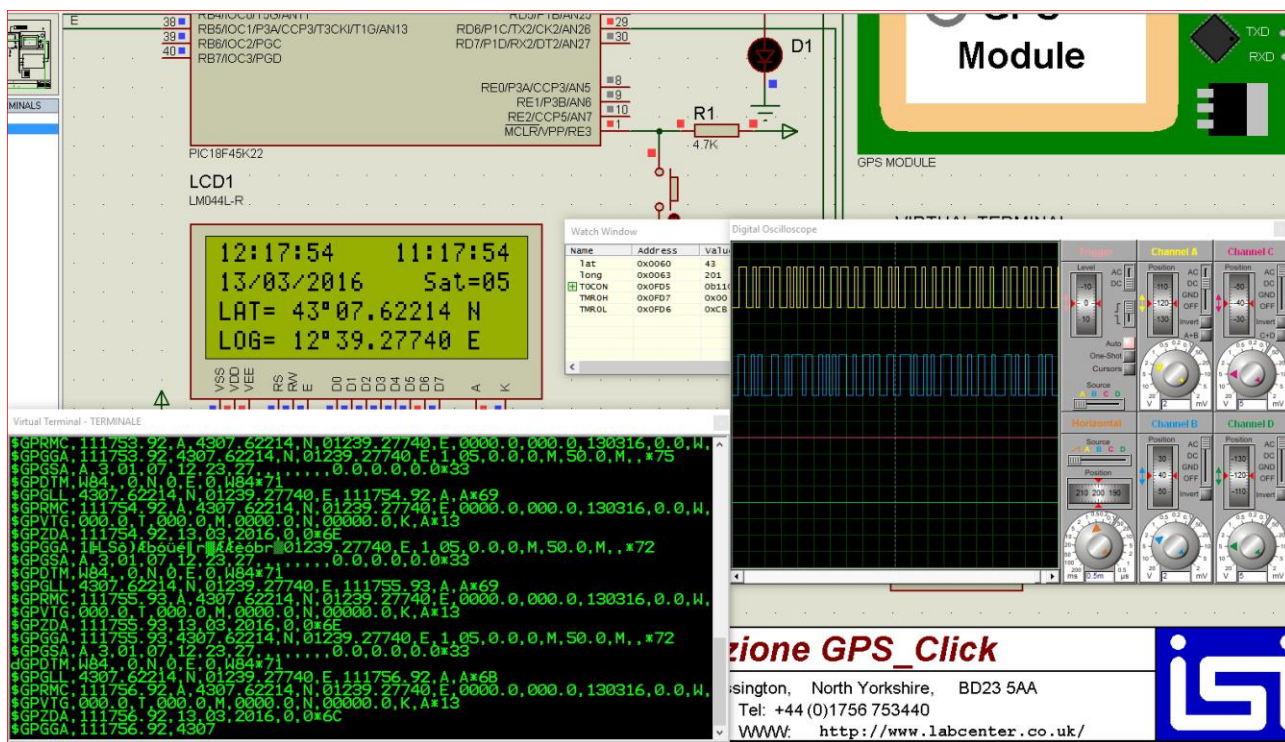



Figura 6: Simulazione.

"string" è possibile individuare la stringa ZDA nel buffer di ricezione(txt) e mediante la funzione memset, vengono azzerati i vettori che dovranno contenere le informazioni relative all'orario e al datario contenute in ZDA. Analogamente a quanto fatto per la stringa GGA,con la funzione memcpy trasferiremo da ZDA ai vettori ore, minuti, secondi, giorno, mese, anno il relativo contenuto informativo, come mostra il listato 3. Ovviamente tutto il software relativo al progetto verrà messo a disposizione del lettore, che potrà scaricarlo dal sito dell'editore.

Osservazioni Finali

Oltre al GPS_click si sarebbe potuto usare, facendo sempre uso dei prodotti della MikroElektronica, il GPS_nano che sfrutta il modulo ricevitore della Hornet.
È il modulo GPS più piccolo in assoluto, con antenna integrata e nonostante le sue dimensioni, offre una sensibilità superiore e prestazioni del tutto ragguardevoli, ad esempio il target può essere individuato con la precisione di 1 m.
Una volta individuate le coordinate geografiche del ricevitore, associando a tali coordinate delle mappe grafiche, ad esempio quelle di Google, si può realizzare un vero e proprio navigatore satellitare. È ovvio che il software di gestione risulterebbe molto più complesso di quello da me presentato, ma ciò non fa perdere di generalità al principio di funzionamento.

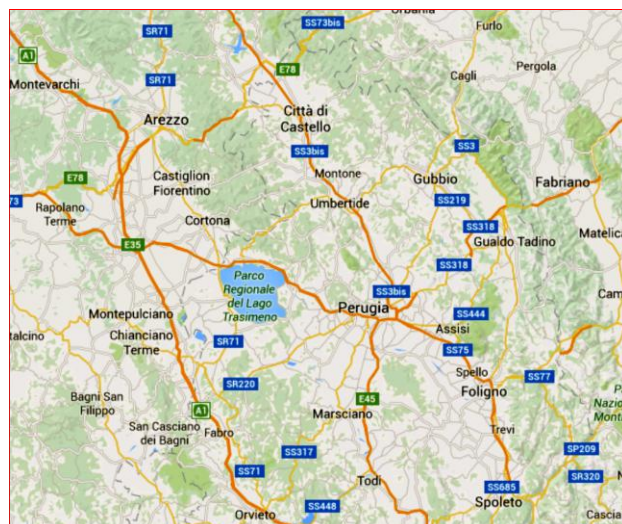



Figura 7

Listato 1

```
void interrupt_high() org 0x0800 iv 0x0008
{
  if (PIR1.RC1IF==1)
   {                                // se generato un interrupt del ricevitore RCIF
     txt[i] = RCREG1;               // memorizza i dati delle stringhe NMEA
     TXREG2=txt[i];                 // invio dato ricevuto GPS su seriale2
     while(PIR3.TX2IF==0)           // aspetto svotamento shift register trasmissione
      {
        asm nop;
      }
     i++;                           // incrementa il puntatore per la ricezione
     if(i ==768)                    //768
      {                             // se sono stati ricevuti tutti i 768 byte
        i = 0;                      // metti i=0
        ready = 1;                  // Pronto per il parsing della stringa GPS
        if(flag_vis==0)
        {
          flag_vis=1;
        }
      }
     PIR1.RC1IF = 0;                // cancella il flag di ricezione
   }
}
```

Listato 2

```
       string = strstr(txt,"$GPGGA");
       if(string != 0)
        { // Se la buffer txt contiene la stringa "$GPGGA" Procediamo...
          if(string[7] != ',')
           {
             //--------- cancello gli array latitudine-longitudine----//
             memcpy(latitudine,string+17,13);  // string+17=inizio latitudine
             memcpy(longitudine,string+30,13); // string+30=inizio longitudine
             //---------- Calcola latitudine-longitudine gradi ------------
             if(latitudine[11]=='S')
              {

                 latitude=(latitudine[0]-48)*10+(latitudine[1]-48);
                 latitude*-1;
              }
             else
              {
                 latitude=(latitudine[0]-48)*10+(latitudine[1]-48);
              }
             if(longitudine[12]=='W')
              {
                   longitude=(longitudine[0]-48)*100+(longitudine[1]-
48)*10+(longitudine[2]-48);
                   longitude*-1;
              }
             else
              {
                  longitude=(longitudine[0]-48)*100+(longitudine[1]-
48)*10+(longitudine[2]-48);
              }
```

Listato 3

```
        string=strstr(txt,"$GPZDA");
        if(string!=0)
         { // Se la buffer di txt contiene la stringa "$GPZDA" Procediamo...
           if(string[7]!=',')
            {
              //---- cancello array orario-datario ----//
              memset(ore,0x00,3);
              memset(minuti,0x00,3);
              memset(secondi,0x00,3);
              memset(giorno,0x00,3);
              memset(mese,0x00,3);
              memset(anno,0x00,5);
              //--- copio della stringa $GPZDA  ore-minuti-secondi ---//
              memcpy(ore,string+7,2);            // string+7=inizio ore
              memcpy(minuti,string+9,2);         // string+9=inizio minuti
              memcpy(secondi,string+11,2);       // string+9=inizio secondi
              memcpy(giorno,string+17,3);        // string+17=inizio giorno
              memcpy(mese,string+20,3);          // string+20=inizio mese
              memcpy(anno,string+23,5);          // string+23=inizio anno;
```

Bibliografia e riferimenti web

http://www.microchip.com/wwwproducts/en/PIC18F45K22
http://www.mikroe.com/easypic/
http://www.mikroe.com/click/gps/
http://www.mikroe.com/click/nano-gps/
http://www.sailsoft.nl/ais_simulator.html
http://www.virtual-serial-port.org/products/vspdxp/?gclid=CPKy6MC5qc0CFckV0wodXQMJAQ

Link di YouTube:

https://www.youtube.com/watch?v=2IlhKohEe90 (Funzionamento reale del sistema)
https://www.youtube.com/watch?v=fFudHuDWZsk (Simulazione con Proteus 7.1)

Figura 8: Attesa triangolazione

Figura 9: Ricezione avvenuta

I primi MCU PIC32 con Core Independent Peripheral

Come superare le limitazioni di costo, potenza e dimensione grazie agli MCU PIC32MM

Quali primi microcontroller PIC32 a disporre delle Core Independent Peripheral, la famiglia PIC32MM offre un conveniente controllo embedded di basso consumo per applicazioni IoT, Consumer, Industriali e BLDC sensorless.

Le Core Independent Peripheral, quali Configurable Logic Cell (CLC) e Multipleoutput Capture Compare PWM (MCCP), spostano compiti dalla CPU, liberandola, per offrire un minore consumo di potenza ed una minore complessità di progettazione. Ulteriori risparmi di potenza, derivanti dalle modalità sleep di basso consumo, si abbinano a piccole opzioni di package 4x4mm nel supportare una maggiore durata delle batterie persino in applicazioni con limitazioni di ingombro.

www.microchip.com/EUPIC32MM

SISTEMA DI MONITORAGGIO GRAFICO DELLA PRESSIONE ATMOSFERICA PER PREVISIONI METEO

Ing. G. Carrera

Si presenta un progetto di applicazione del display Nokia 5110, esso rappresenta in formato grafico l'andamento della pressione atmosferica e, a richiesta, stampa sul display in formato testo la temperatura, la pressione e la quota altimetrica. Il trend della pressione permette una facile previsione meteo a breve termine.

Questo piccolo ed economico display grafico da 84x48 pixel ha una diagonale di soli 1,5" ma ha una buona
visibilità anche grazie a una retro-illuminazione di 4 led blu, disposti sui due lati corti. Questo LCD è stato usato nei vecchi telefoni cellulari della Nokia (modelli 3310 e 5110) e viene ora prodotto come compatto display per sistemi a microcontrollore, montato su una piccola scheda munita di 8 contatti strip. Sul modulo è montato un controller grafico PCD8544 della Philips. Per Arduino sono disponibili in rete diverse librerie, quella che verrà usata è di Adafruit, che fornisce anche la scheda insieme a numerosi altri produttori. Va fatta attenzione solo al layout e alla disposizione dei contatti, che sono diversi a seconda del costruttore. L'interfaccia con il microcontrollore richiede 4 o 5 segnali, i dati sono trasmessi via bus SPI unidirezionale (MOSI e SCK). La libreria prevede di usare il bus SPI hardware di Arduino, soluzione più veloce, oppure di utilizzare qualsiasi altro pin con un'interfaccia software, più lenta. Nel progetto proposto è stata usata la prima soluzione, la tabella seguente illustra i collegamenti, sia che venga usato un sistema Arduino Pro Mini da 3,3 V, sia nel caso si utilizzi un chip Atmega328P con boot-loader, come quello usato nel prototipo. La scelta dell'alimentazione della CPU a 3,3 V deriva dal fatto che le alimentazioni e i segnali dei moduli del display e del sensore utilizzato sono a 3,3V (da 2.7 a 3.3 V). Questa scelta non è obbligata in quanto esistono dei chip adattatori di livello, ma è senz'altro la soluzione più semplice e compatta. Oltre ai due fili del bus SPI, sono utilizzati

Figura 1: Il display Nokia 5110.

Collegamenti display LCD

Arduino pin	CPU Pin #	CPU segnale	Funzione LCD	LCD Pin#
D3	5	PD3	RST	1
D4	6	PD4	CS	2
D5	11	PD5	D/C	3
D11	17	PB3	DIN (MOSI)	4
D13	19	PB5	CLC (SCK)	5
			Vcc (+3,3 V)	6
			BL (Backlight Led)	7
			Gnd	8

Tabella 1

altri tre segnali: il Reset (RST=0), il Chip select (CS=0) e il Mode select (D/C: 1 = Data mode e 0 = Command mode). Tutti i segnali sono configurati come uscite, se si usasse un Arduino a 5 V, sarebbe necessario usare cinque adattatori da 5 V a 3,3 V, ad esempio un CD4050 alimentato a 3,3 V che, a differenza di altre porte logiche, tollera ingressi a tensioni maggiori rispetto alla sua alimentazione. Il segnale BL corrisponde agli anodi dei led di retroilluminazione. Si può regolare la luce dei led usando un bit PWM, con opportuno resistore limitatore oppure con luce fissa. Si è scelta questa soluzione alimentando i led con i 5 V d'ingresso con un resistore di 220 Ω. In questo caso si riduce la corrente di alimentazione dell'alimentatore a 3,3 V e si semplifica il sistema.

Il sensore BMP85

Questo trasduttore digitale, prodotto dalla Bosch, misura la pressione con un range di 300-1100 [hPa] con la risoluzione di 0,01 [hPa] e con un'accuratezza tipica di ±0,2 hPa. Nel Sistema Internazionale un hPa è pari a un millibar. Il BMP85 si presta sia per misurare la pressione atmosferica sia l'altitudine, che dipende prevalentemente da essa, con un range da +9000 a -500 [m]. Esso incorpora anche un sensore di temperatura con una risoluzione di 0,1 [°C] con un'accuratezza tipica di

Arduino pin	CPU Pin #	CPU segnale	Funzione BMP85	scheda Pin#
			Gnd	1
			EOC	2
			XCLR	3
A5	28	PC5	SCL	4
A4	27	PC4	SDA	5
			Vcc (+3,3 V)	6

Tabella 2

±0,5 °C. La comunicazione con il microcontrollore avviene con bus I2C di cui il chip Atmega328 è dotato. Anche di questo sensore sono disponibili librerie per Arduino e sono fornite da vari produttori delle piccole schede (breakout board) come quella usata. I collegamenti di questo sensore con Arduino richiedono solo i due segnali SDA e SCL, come visibile dalla tabella 2. Il segnale di input XCLR (master clear) serve per resettare e inizializzare il chip, cosa che avviene anche quando si dà alimentazione, se non usato va lasciato libero. Il pin di output EOC (End Of Conversion) può essere usato per vedere se la conversione AD è terminata, anche questo segnale non è usato. Il BMP085 è uno smart sensor, ossia un trasduttore con uscita digitale che fornisce i parametri di calibrazione in modo da ricostruire le misure in grandezza fisica ed evitarci una difficile calibrazione. I dati di pressione e di temperatura, infatti, vanno compensati con i coefficienti di calibrazione contenuti nella E2PROM del BMP085. Ogni senso-

Figura 2: La scheda del sensore.

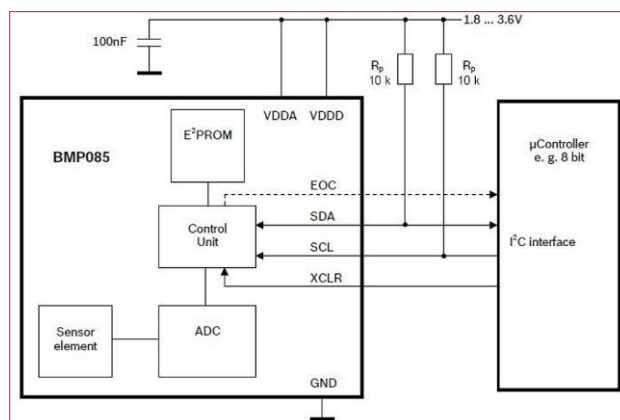



Figura 3: Schema funzionale del chip BMP085

re ha immagazzinati nella E2PROM i suoi 11 coefficienti di calibrazione, a 16 bit, che permettono di calcolare temperatura e pressione. Il software deve prima leggere questi coefficienti, quindi leggere i dati non compensati di temperatura (16 bit) e pressione (da 16 ia 19 bit) e poi calcolare temperatura e pressione compensate e in grandezza fisica. Il costruttore fornisce anche il codice di calcolo da cui sono state ricavate le librerie per Arduino. Per maggiori dettagli si rimanda alla bibliografia.

Calcolo dell'altitudine

Supponendo che la temperatura diminuisca con la quota secondo un gradiente costante:

$$T(h) = T_0 - a \cdot h$$

La formula ipsometrica che tiene conto di questo gradiente è indicata in tavola 1. Dove, esprimendo le grandezze fisiche nel Sistema Internazionale:

p: la pressione misurata [hPa]:
p0: pressione al livello del mare=1013,25 [hPa]:
a: gradiente standard di temperatura = 0,0065 [K/m]:
M: massa molare dell'aria = 0,0289644 [kg/mol]:
g: accelerazione di gravità = 9,807 [m/s2]:
R: constante universale dei gas = 8,3143 [N*m/(mol*K)]:
T0: temperatura standard (+15°C) in gradi kelvin = 288,15 [K]:
h: altitudine sul livello del mare [m].

$$p = p_0 \cdot \left(1 - \frac{a \cdot h}{T_0}\right)^{\frac{M \cdot g}{R \cdot a}} = p_0 \cdot \left(1 - \frac{a \cdot h}{T_0}\right)^{5.2553}$$

Formula 1

$$h = \frac{T_0}{a} \cdot \left(1 - \frac{p}{p_0}\right)^{\frac{R \cdot a}{M \cdot g}} = 44330.77 \cdot \left(1 - \frac{p}{p_0}\right)^{\frac{1}{5.2553}}$$

Formula 2

Ricavando l'altitudine dalla formula precedente si ha la formula di tavola 2. Questa è anche la formula suggerita nel data sheet della Bosch, per cui, con una variazione di pressione di 1 hPa si ha una variazione di quota di 8,43 m a livello del mare. Questa formula è anche quella usata dalla libreria Adafruit-BMP085-Library-master, ma è sempre meglio indagare da dove vengono i numeri e le ipotesi che sono state fatte. Facendo una opportuna media delle misure si arriva, secondo il costruttore, ad un rumore inferiore a 0,1 m. La figura 4 mostra l'altitudine in funzione della pressione atmosferica. Una maggiore accuratezza si potrebbe ottenere conoscendo la vera pressione atmosferica p0 al livello del mare, che dipenderà dalle condizioni atmosferiche. Va comunque detto che non esiste formula in grado di modellare l'atmosfera terrestre, soprattutto la troposfera, zona più vicino alla terra che risente fortemente delle perturbazioni meteo. La densità dell'aria, quindi anche la pressione, dipende dalla temperatura e dalla quantità di vapore che diminuiscono con l'aumentare della quota. Le molecole dell'acqua (peso mole-

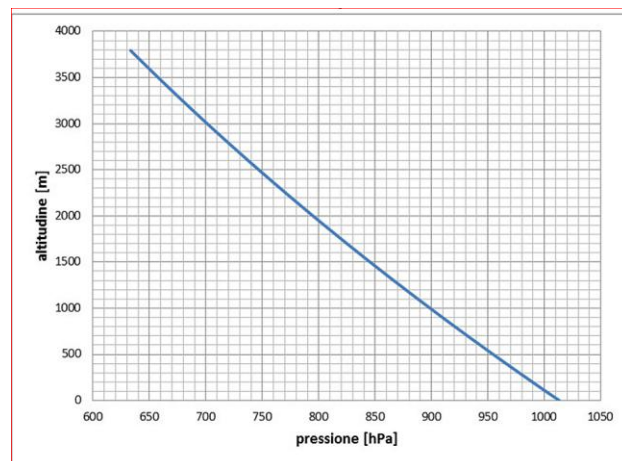



Figura 4: Relazione tra pressione e quota.

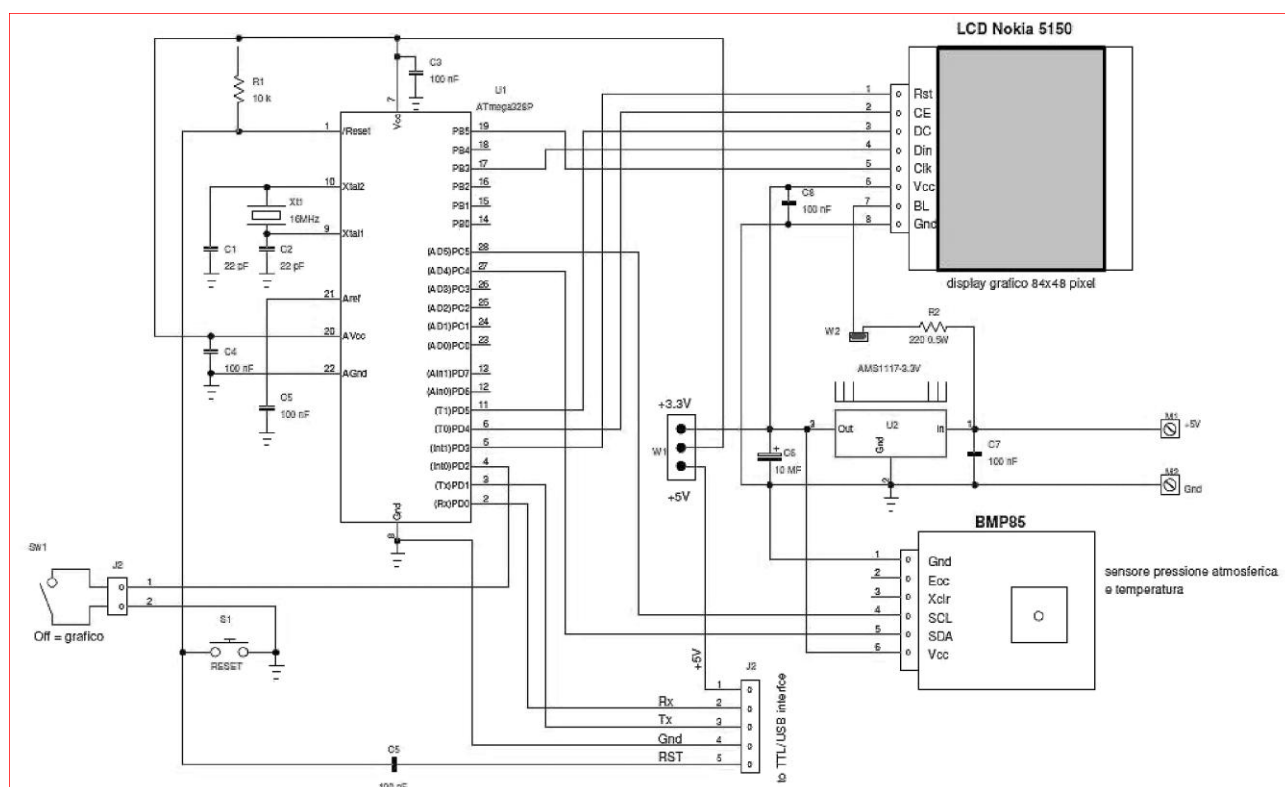



Figura 5: Schema elettrico.

colare = 18 unità di massa atomica) tendono a sostituirsi a quelle dell'azoto, più pesante (28 u.m.a.), per cui l'aria umida è più leggera e produce bassa
pressione. Al contrario, l'aria secca è più pesante e produce alta pressione. Questo spiega il perché le misure altimetriche con la pressione non sono sempre così precise. L'altitudine è oggi misurata anche dai ricevitori GPS, ma la misura non sempre coincide con l'altitudine vera, questo per varie cause tra cui la diluizione geometrica VDOP e gli errori del sistema. Gli aerei usano anche uno speciale radar altimetrico, che permette di ricavare la distanza dal suolo sottostante che coincide con l'altitudine solo se l'aereo vola sul mare, e di rilevare monti o altri ostacoli.

Lo schema elettrico

La figura 5 mostra lo schema elettrico del sistema proposto. Io ho usato un chip Atmega328P programmato con il bootloader Arduino e alimentato a 3,3 V. Con questa alimentazione il quarzo dovrebbe essere da 8 MHz, così come consigliato dalla Atmel, ma già in diversi sistemi ho utilizzato quarzi da 16 MHz senza problemi. Per non avere problemi è meglio usare il quarzo da 8 MHz. Preferisco usare questa alimentazione perché usata sia dal display LCD sia dal sensore di pressione. Un'altra soluzione è quella di impiegare una scheda Arduino Pro Mini nella versione a 3,3V, in figura 6 si può vedere lo schema di questa alternativa. Tutti e tre i moduli e l'alimentatore da 3,3V possono essere montati su una scheda millefori e collegati con connettori strip. È preferibile programmare la scheda Arduino prima di inserirla nel sistema. È facile reperire in commercio anche una breakout board per l'alimentatore a 3,3V, spesso munita di connettore micro USB per usare un comune alimentatore a 5V dei più recenti cellulari. In realtà Arduino Pro Mini 3.3 ha già incorporato un piccolo alimentatore, ma esso deve

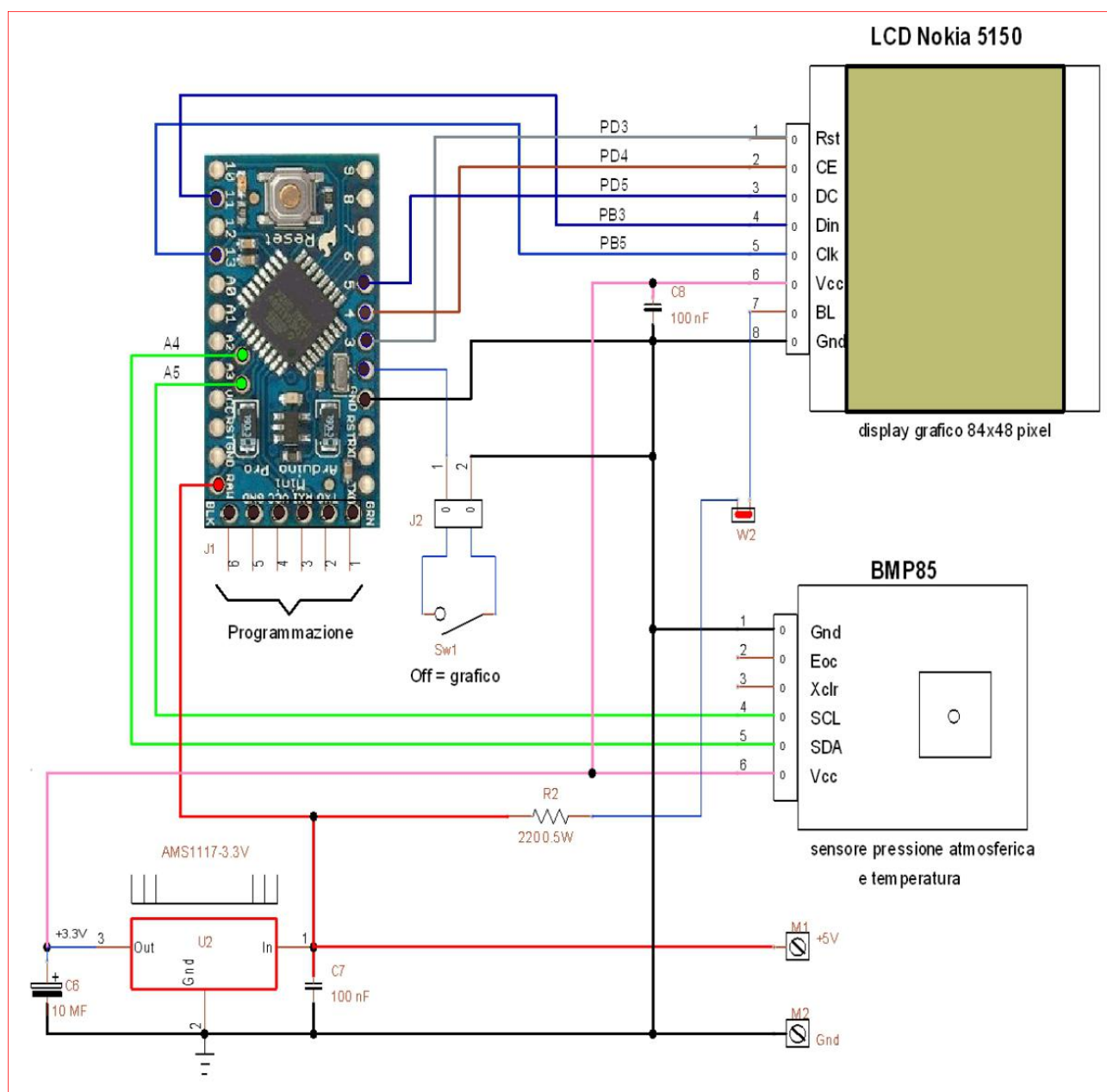



Figura 6: Schema della variante con Arduino Pro Mini a 3,3V.

alimentare il chip Atmega328 e non ha una superficie dissipatrice sufficiente per fornire corrente anche al nostro display, specie per tempi molto lunghi. Per cui ho preferito usare un AMS1117 saldato su una piastrina di rame di circa 250 mm².

Il prototipo

La scheda madre è una millefori, il display e il sensori sono connessi con connettori strip passo 0,1” da 8 e 6 pin. Nella figura seguente si puè vedere il prototipo in funzione. Un secondo interruttore o jumper w2 serve per togliere la retroilluminazione, riducendo sensibilmente il consumo. Avendo il regolatore AMS1117 usato da me un case smd tipo SOT-223, l’ho saldato sotto, sul lato saldature della piastra millefori.

Il programmatore

La piccola scheda a sinistra, in figura 7, serve per la programmazione e contiene un adattatore seriale USB e un alimentatore da 5V a 3,3V. Essa serve anche come interfaccia seriale USB, ma in questo caso il suo uso prevalente è quello di trasferire

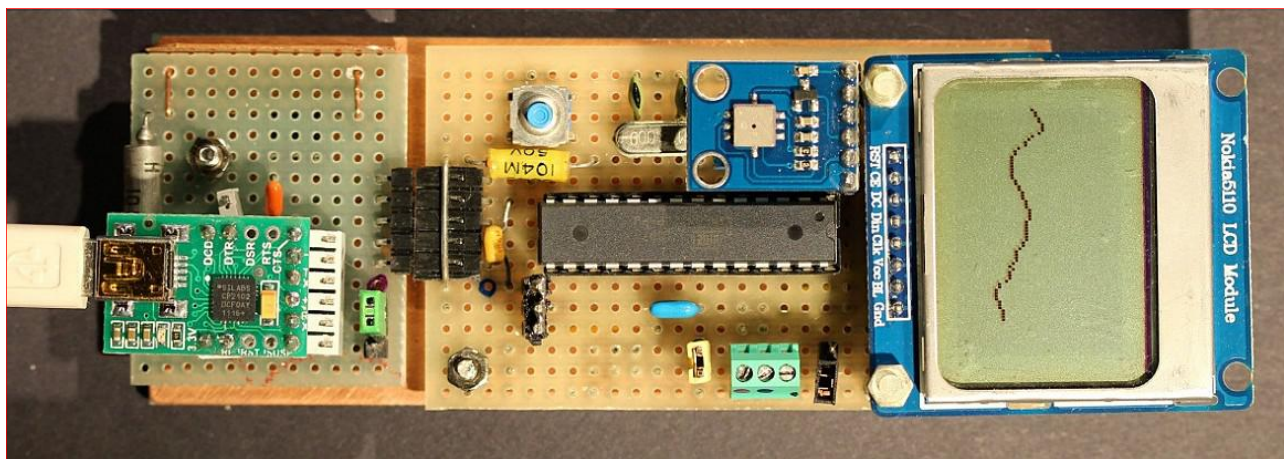



Figura 7: Il prototipo in funzione.

lo sketch, compilato dalla IDE di Arduino, al nostro sistema. Essa è in grado di programmare il chip a 3.3V, ma anche a 5V, spostando un ponticello, nel nostro caso va impostato a 3.3V. Nella IDE di Arduino si deve impostare la scheda come Arduino Pro a 16MHz a 5V (nel mio caso, anche se alimentata a 3,3V) oppure a 8 MHz 3,3V a seconda del quarzo usato. Una volta programmato, se tutto va bene, si può rimuovere la scheda di programmazione e alimentare la scheda dalla morsettiera. Per mia comodità, visto le numerose modifiche software, ho preferito alimentare il sistema dalla scheda di programmazione, anche durante il normale funzionamento, per cui ho cortocircuitato i tre pin del connettore w1 di figura 5 con un connettore femmina. Se servisse la seriale per inviare le misure anche a

Figura 8: Programmatore seriale.

un PC, si può lasciare la scheda del programmatore e non montare l'alimentatore. Per programmare a 3,3V occorre anche un piccolo alimentatore da 5V a 3,3V e si deve fare attenzione che i livelli dell'adattatore siano anch'essi a 3,3V. In commercio si possono facilmente trovare questi programmatori adatti sia per programmare a 5V sia a 3,3V spostando un ponticello. Io, come al solito, ho preferito farlo usando un adattatore con chip CP2102. Lo schema di figura 8 mostra i collegamenti. Il connettore, del tipo strip maschio a 5 pin a 90°, è adatto per il mio prototipo. Per chi usasse la scheda Arduino Pro Mini, occorre che utilizzi un programmatore specifico o che cambi la disposizione dei segnali sul connettore che ora è a 6 pin. Naturalmente occorre anche caricare il driver dell'adattatore, che deve avere, oltre ai segnali Rx e Tx, anche il DTR che serve per resettare il sistema.

Gli screenshot

Le immagini di figura 9 mostrano i tre possibili stati visualizzati sul display, il primo è il logo che ho creato con un programma di grafica tipo Windows Paint e convertito il file bitmap in testo con il programma LCD Assistant (http://en.rad-

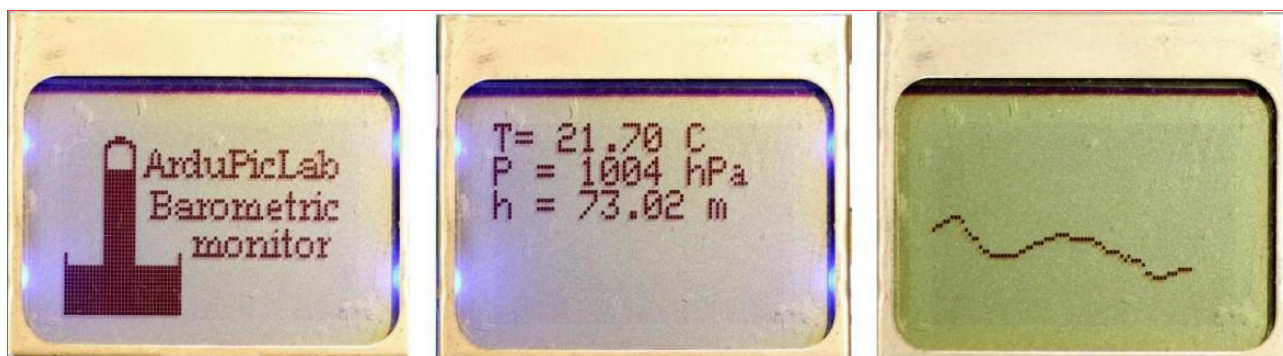



Figura 9: Gli screenshot: a = logo, b= misure, c = grafico pressione.

zio.dxp.pl/bitmap_converter/). Ho cercato di disegnare, nonostante la bassa risoluzione del display, il barometro di Torricelli. Dopo aver convertito l'immagine in una serie di 84*48/8= 504 caratteri hex, si deve sostituirla nel file di libreria Adafruit_PCD8544.cpp a partire dalla riga 42, del vettore pcd8544_buffer che è proprio la memoria grafica del display, che viene mostrata con la funzione display.display() e azzerata con display.clearDisplay(). Il logo è mostrato al momento dell'accensione per qualche secondo. Se si vuole usare il mio logo, dopo il listato del programma sono disponibili i valori iniziali del buffer del display che sono visualizzati in fase di inizializzazione. La seconda schermata è quella delle misure, la terza riguarda il grafico della pressione, che nel caso mostrato denota condizioni meteo variabili verso bassa pressione. Si noti che la registrazione non è ancora arrivata a riempire il buffer e che i led di retroilluminazione sono stati spenti (jumper w2 aperto) per risparmio energetico, come si vede anche in figura 7.

Il programma

Esistono due intervalli di tempo, uno corto o periodo base che è deltat = 3 secondi e uno lungo che è il periodo di campionamento vero e proprio ed è il multiplo tbmult del precedente, quindi:

Ts = tbmult*deltat

Un interruttore o un pulsante stabilisce se il sistema deve fare il grafico della pressione (sw1 aperto) o stampare in formato alfanumerico le tre misure, come si vede nelle figure 9c e 9b. Per generare i tempi non ho usato la solita delay() ma la funzione millis() che restituisce il tempo in millisecondi dall'accensione, usando una scala temporale assoluta il risultato è più preciso. Nel caso di sw1 chiuso, ogni periodo base il sistema legge la pressione atmosferica e la temperatura, calcola l'altitudine con la formula (1) e stampa le tre misure in formato alfanumerico: p [hPa], T [°C] e h [m]. Durante tutto il tempo che sw1 è chiuso non sono abilitate le acquisizioni per la grafica. Nel caso di sw1 aperto il sistema legge la pressione atmosferica, calcola la differenza e la scala nel formato del display e memorizza il dato in un buffer che ha la stessa dimensione dell'asse x del display, per cui l'intervallo di tempo Tr su cui si visualizza il grafico delle variazioni di pressione atmosferica rispetto al valore iniziale sarà pari a:

Tr = Ts*84

Se tbmult=400, il periodo di campionamento è Ts = 1200 secondi = 1/3 ore, per cui Tr = 84x1/3 = 28 ore, poco più di un giorno. Con

tbmult=100 si ha Ts = 300 secondi con Tr = 7 ore, e ho visto che è sufficiente per una previsione meteo. Il display ha una risoluzione verticale di solo 48 pixel, quindi è necessario eseguire alcune elaborazioni:

1. ricavare le differenze di pressione pdiff, rispetto al primo valore misurato:
2. adattare i pdiff di ±960 Pa ai valori 0 e 47 del display 5110;
3. fare il complemento del numero a 47 per avere lo zero in basso a sinistra invece che in alto.

Con un pdiff iniziale sempre uguale a zero, il primo punto è a metà della prima colonna a destra. La prima elaborazione riduce le cifre che rappresentano la pressione atmosferica, espressa in pascal, da 5 cifre a solo tre.
Questo perché l'escursione della pressione in un giorno raramente supera i 9 hPa, questo l'ho verificato personalmente su misure effettuate in alcuni anni dalla mia stazione meteo. La seconda e terza elaborazione sono fatte agevolmente dalla potente funzione Arduino:

```
map(pdiff, -960, +960, 47, 0);
```

Infatti, essa fa corrispondere il minimo della pdiff (-960) a 47, che è il punto più basso dello schermo. Se l'accuratezza dichiarata dal costruttore è di ±0,2 hPa, occorre avere una risoluzione doppia, per cui si ha 0,4 hPa x 24 punti = 960 Pa, ossia un range ± 9,6 hPa sui 48 pixel verticali. Nella prima scansione i punti delle misure sono visualizzati a iniziare dal lato sinistro (x=0), aggiungendo un nuovo punto ogni Ts minuti e mettendo i dati, in formato adattato al display nel buffer delle misure data[84]. Una volta arrivati sull'estremità destra, all'ascissa x=83, cambia il modo di rappresentazione, tutti i punti scorrono di una posizione verso sinistra e l'ultimo valore viene messo sull'estremità destra.
A ogni intervallo di campionamento è scartato il contenuto del data[0] che assume il valore del data[1] e così via, per cui: data[j]=data[j+1]. Il nuovo dato lastv è scritto nel data[83]. Lo shift dei dati verso sinistra crea uno scorrimento della traccia con un effetto simile a quello dei vecchi registratori a carta e si ottiene un trend storico della pressione nel periodo Tr, assai utile per le previsioni meteo. Per questo programma sono state usate le librerie Adafruit [1] e [2] per il display 5110 e la [3] per il sensore BMP085.

Librerie usate

1) Adafruit-PCD8544-Nokia-5110-LCD-library-master
2) Adafruit-GFX-Library-master
3) Adafruit-BMP085-Library-master

Bibliografia

1) "Nokia 5110/3310 Monochrome LCD", Adafruit Industries, 08 Sept 2015.
2) "PCD8544 48x84 pixels matrix LCD controller/driver", Philips Semiconductors, 12 Apr 1999.
3) "BMP085 Data sheet" rev.1, Bosch Sensortec, 01 July 2008.
4) "On the barometric formula", Mario N. Berberan-Santosa, Evgeny N. Bodunov, Lionello Pogliani, American Association of Physics Teachers, 1997.

BUGGY CAR

di Moreno Palazzo

Un concentrato di tecnologia su 4 ruote motrici realizzato dalla MikroElettronica.

Finalmente una piattaforma non convenzionale con cui gli appassionati di elettronica possono testare i loro progetti o programmi. Volendo rompere gli schemi, credo che la MikroElektronika abbia fatto nuovamente centro, creando un prodotto sì differente ma nel contempo utile ed innovativo, di fatti non si tratta solo di una demoboard standard composta da led, potenziometri, pulsanti, display ecc.. Al contrario la BUGGY è dotata di diversi elementi che costituiscono nell'insieme un concentrato di tecnologia, rendendolo anche “leggero” per la sua forma che ricorda un'autovettura. Proprio per questa forma di leggerezza, si può utilizzare BUGGY a scopo didattico o semplicemente utilizzata da mani esperte, si possono creare progetti di automotive e non, molto complessi. Una delle svariate caratteristiche è quella di avere tutto il necessario che un progettista possa desiderare, dalla batteria a litio, alla possibilità di poter controllare BUGGY con qualsivoglia tipo di MCU. Dato che a bordo non vi è nessun tipo di microprocessore, l'utilizzatore finale resta libero di scegliere se acquistare i prodotti della gamma MIKROE o di adattare qualsiasi tipo di MCU più consona alle proprie esigenze, proprio per questo motivo è stata dotata di diversi I/O corrispondenti alle varie sezioni del C.S.
Di due driver per motori in CC, led e tre schede dette MicroBus Sockets click, sfruttando le connessioni SPI, I2C, Analog, UART, Interrupt, PWM, Reset and Power supply, sarebbe possibile con un numero di 100 click boards, dar sfogo alla fantasia. Analizzando la figura 1, vediamo uno spaccato in tre proporzioni di tutte le parti e componenti appena citati. Nella figura 2, vi sono riportate tutte le connessioni dei pin I/O della piattaforma, quindi risulta

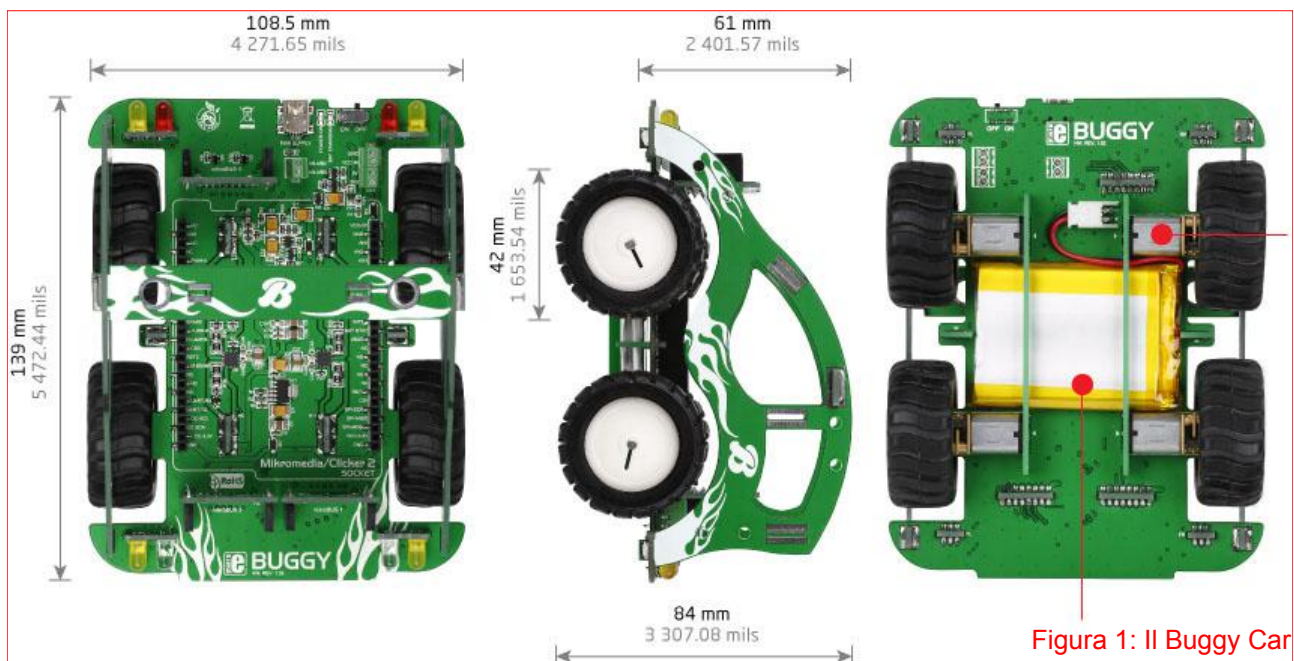



Figura 1: Il Buggy Car

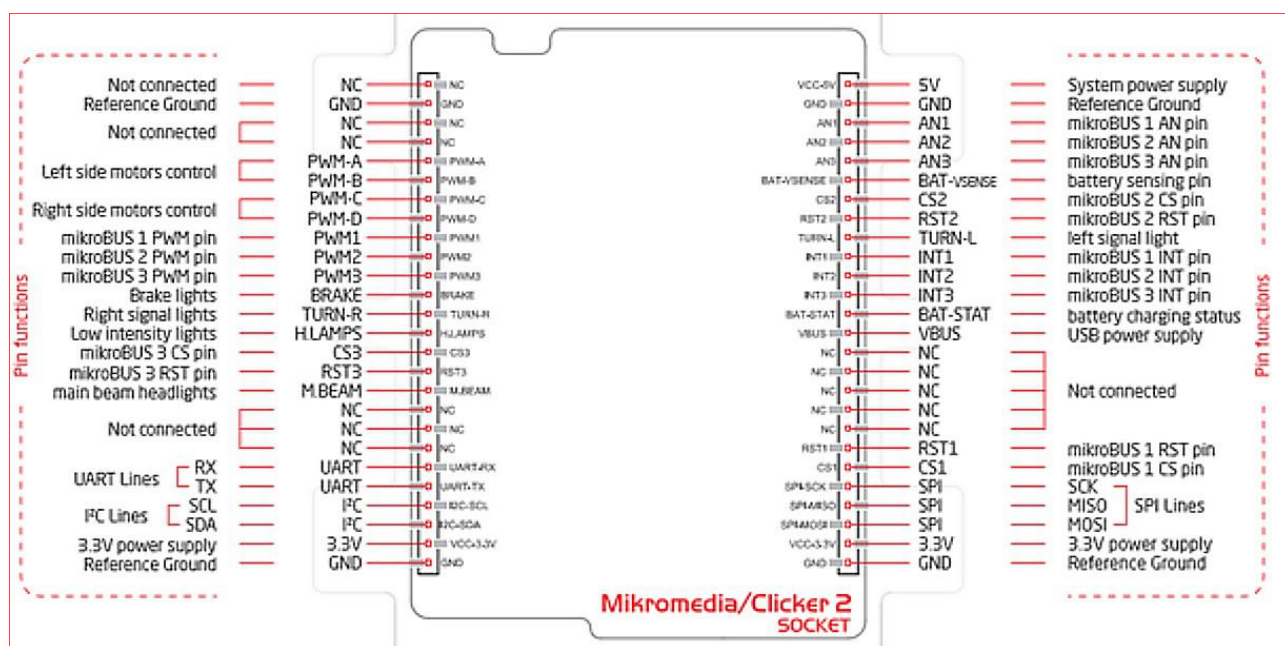



Figura 2: Le connessioni dei pin I/O della piattaforma,

molto facile adattare un progetto pensato con un Raspberry Pi o un Arduino, semplicemente creando delle connessioni tra la piattaforma e il vostro dispositivo. Dopo aver assemblato la struttura ho testato con un cablaggio volante tutte le connessioni studiando il PDF fornito dalla casa costruttrice, eseguendo un link tra l'alimentazione ed una connessione I/O, come per esempio il PWM-A le ruote corrispondenti si attivano, ripetendo l'operazione anche per i led vedremo che questi ultimi si accendono.

Figura 3: PCB

Ciò che rende potente BUGGY è il fatto che non solo si può adottare ogni sorta di MCU, ma sia i driver per motori in CC che i led, funzionano con l'alimentazione a 5V o 3.3V, essendo dotati di transistor, quindi volendo risparmiare ma nel contempo divertirsi, ho provveduto alla realizzazione di un C.S.

Con l'implementazione del famigerato PIC16f876, un microprocessore dalle ridotte dimensioni, ma dalle caratteristiche estremamente esaustive, scelto non per caso. Prendendo come riferimento le dimensioni della piattaforma per mezzo di un calibro, ho sviluppato un disegno cad interfacciando il microprocessore alle connessioni, cercando di sfruttare tutte le periferiche disponibili secondo il pic selezionato, questo perché ognuno di noi può sentirsi veramente libero di scegliere se realizzare un progetto motorizzato, includendo il proprio linguaggio di programmazione che non sia per forza della MikroE.

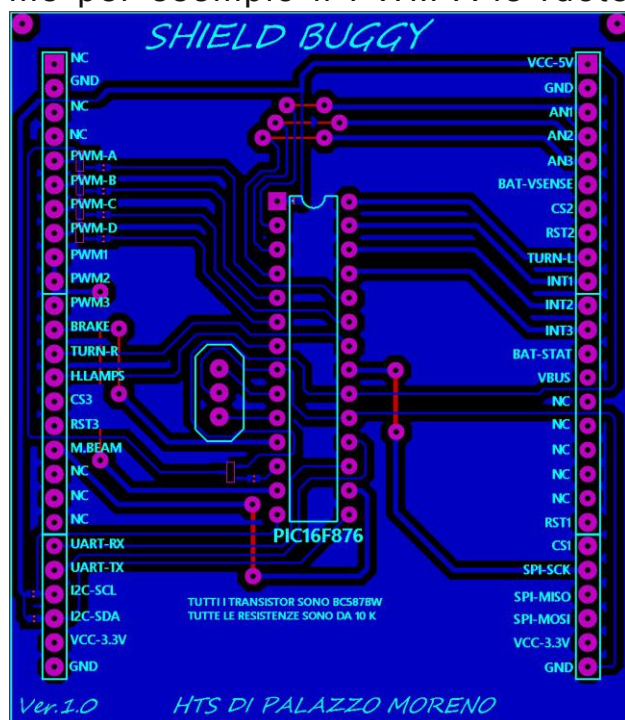
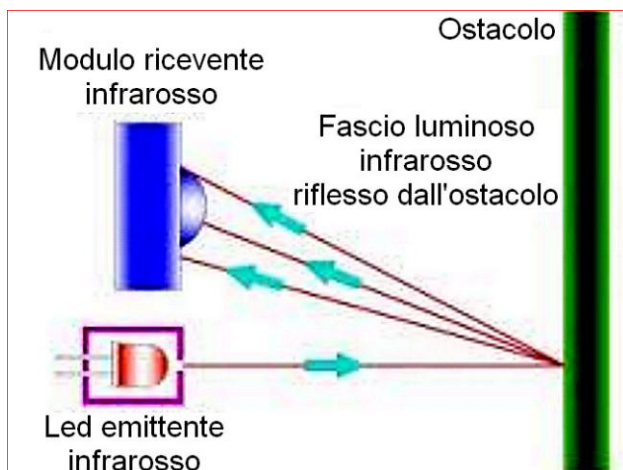



Figura 4: Esempio con infrarosso.

Realizzazione pratica

Come si evince dalla figura 3 la scheda è stata progettata sulla base della piattaforma riportando tutte le connessioni.

Ciò che non risalta subito, ma di fondamentale importanza, sono i transitor BC587BW e delle resistenze smd, i transitor utilizzati in questo caso pilotano gli ingressi PWM-A, PWM-B, PWM-C e PWM-D.

Naturlamente avrei potuto anche non metterli e decidere di collegare direttamente gli ingressi citati al Pic, ma dato che lo stesso è fornito di due porte CCP1 e CCP2 tramite le funzioni MikroBasc, si possono creare due pwm distinti,

Figura 5: Realizzazione pratica.

uno dei quali è stato proprio interconesso, per mezzo di un transitor, ai collettori dei transitor relativi ai PMW-A/B/C/D che, a loro volta vengono pilotati da altre 4 periferiche del Pic.

Tutto questo perché i PWM1/2, sono regolabili con il mino sforzo e viene utilizzato come acceleratore, potendo, appunto, variare il duty-cicle. Il PWM2 invece, senza transistor alcuno è connesso ai rispettivi PWM1/2/3, naturalmente si può utilizzare questa funzione per pilotare dei led IR emittitori, realizzando un “baffo” infrarosso sfruttando la rifrazione della luce IR, vedi figura 4.

Una delle molteplici applicazioni però, potrebbe anche essere la creazione di un controller IR e un ricevitore montato a bordo. Come tralasciare anche le potenti porte analogiche?

Di fatti non è possibile, perché un “ROBOT” che si rispetti ha in dotazione almeno un sensore di temperatura, uno per la luce e molti altri, per cui gli I/O AN1/2/3 sono collegati al Pic, proprio come i pin inerenti ai led direzionali, di stop che d'illuminazione frontale; dando uno sguardo alla figura 5 vediamo come rende l'idea il progetto iniziale. In conclusione credo che il limite a questo punto possa esser solo il tempo e la fantasia a nostra disposizione.

Programmiamo il PIC

Volendo creare un semplice ma funzionale programmino in Basic, come da buona norma è necessario realizzare il classico “HELLO WORD!” (frase che venne utilizzata durante l'accensione del primo pc, tradizione che viene ancora oggi utilizzata da molti progettisti come segno di buon auspicio) pertanto il nostro primo

Listato 1

```
' * Project name:
'     BUGGY ("HELLO WORD!")
' * Copyright:
'     (c) HTS di PALAZZO MORENO, 2016
' * Description:
'     Questo programma utilizza le porte CCP del pic, come PWM1/2 alcune periferiche
'     per l'attivazione degli ingressi PWM-A/B/C/D, dei led e una serie di azioni
'     di test per il moto della piattaforma BUGGY.
' * Test configuration:
'     MCU:             PIC16F876
'     Dev.Board:       BUGGY
'     Oscillator:      HS, 04.0000 MHz
'     Ext. Modules:    - SHIELD V1.0
'     SW:              mikroBasic PRO for PIC
'                      http://www.mikroe.com/mikrobasic/pic/
' * NOTES:
' *
```

Listato 2

```
program buggy
' Dichiarazione dei simboli corrispondenti alle porte I/O
symbol PWMA = portb.3
symbol PWMB = porta.3
symbol PWMC = portb.2
symbol PWMD = porta.4
symbol AN1  = porta.0
symbol AN2  = porta.1
symbol AN3  = porta.2
symbol TURNL = portb.7
symbol TURNR = portb.1
symbol HLAMP = porta.5
symbol BREAK = portc.0
symbol MBEAM = portc.5
```

Listato 3

```
' Variabili di comodo per i PWM
dim current_duty, current_duty1, old_duty, old_duty1 as byte
' Variabili di comodo
dim count, i, j, adc as byte
```

Listato 4

```
sub procedure InitMain()
  PORTA = 0
  TRISA = 255                     ' configurazione della porta come imput
  PORTB = 0                       ' spegni la portb
  TRISB = 255                     ' configurazione della portb come output
  PORTC = 0                       ' spegni la portb
  TRISC = 0                       ' configurazione della portc come output
  PWM1_Init(5000)                 ' inizializza il PWM1 a 5KHz
  PWM2_Init(5000)                 ' inizializza il PWM2 a 5KHz
  PWM1_Start()                    ' start PWM1
  PWM2_Start()                    ' start PWM2
end sub
```

Listato 5

```
main:
  InitMain()
  current_duty1 = 16                 ' setta il valore 16 alla variabile current_duty1
  PWM2_Start()                       ' start PWM2
  PWM2_Set_Duty(current_duty1)       ' assegna il valore current_duty1 per il  PWM2
test:
' Lampeggio dei led direzionali DX SX
  for i = i to 5
    TURNL =  1
    delay_ms(500)
    TURNL =  0
    delay_ms(500)
  next i
  for i = i to 5
    TURNR =  1
    delay_ms(500)
    TURNR =  0
    delay_ms(500)
  next i
' Attivazione comando HLAMP, BREAK, MBEAM
    HLAMP = 1
    delay_ms(5000)
    HLAMP = 0
    delay_ms(5000)
    BREAK = 1
    delay_ms(5000)
    BREAK = 0
    MBEAM = 1
    delay_ms(5000)
    MBEAM = 0
' Attivazione delle rute motrici, marcia avanti 10 sec
    PWMA  =  1
    PWMC  =  1
    delay_ms(10000)
    PWMA  =  0
    PWMC  =  0
' Attivazione delle rute motrici, ruta 360° 5 sec
    PWMB  =  1
    PWMC  =  1
    delay_ms(5000)
    PWMB  =  0
    PWMC  =  0
' Attivazione delle rute motrici, marcia indietro 10 sec
    PWMB  =  1
    PWMD  =  1
    delay_ms(10000)
    PWMB  =  0
    PWMD  =  0
    PWM1_Stop()                       ' stop PWM1
    PWM2_Stop()                       ' stop PWM2
end.
```

programma, andrà ad azionare le ruote motrici e i led come da seguente schema:

- accensione led bianchi;
- accensione led direzionali;
- accensione led di stop;
- attivazione PWM-A/B/C/D;
- avanti 10 sec;
- stop;
- ruota 360°;
- avanti 10 sec;
- stop.

Scriviamo il programmino in MikroBasic e rendiamolo il più professionale possibile, vediamo come. La prima regola è quella di scrivere, a inizio programma, tutte le caratteristiche e funzioni, per renderlo intuibile già dal primo sguardo, come da listato 1.
Tutto quello che viene scritto dopo la virgoletta il compilatore non lo considera, essendo commenti vengono ignorati, adesso andremo ad analizzare tutte le parti del programma HELLO WORD.
In primo luogo, dopo aver scritto tutte le caratteristiche, troviamo il nome del programma, nelle fasi successive, seguendo uno schema ben preciso dettato dal compilatore, andremo a scrivere tutti i symbol, variabili, sub procedure, l'inizializzazione dei registri e variabili, le label contenenti i "part program" ed infine il comando END, che determina la fine del programma.
Quello che abbiamo scritto nella dichiarazione dei SYMBOL, vedi listato 2, altro non è che l'assegnazione delle voci corrispondenti alle I/O della piattaforma, associandoli agli ingressi del PIC, naturalmente per una comodità tecnica e per un eventuale modifica semplice e veloce. Andiamo a scrivere le variabili circa i due PWM e quelle di comodo, consiglio sempre d'inserirle per poi utilizzarle al momento opportuno (listato 3).

A seguire andremo a scrivere le SubProcedure, come nel caso di specie, la Sub InitMain() che inizializza i registri del pic, come mostrato nel listato 4.
Con le label MAIN e TEST, andiamo a richiamare l'inizializzazione del Pic, settiamo la variabile "current_duty1" e attiviamo il PWM2, altresì con la label TEST finalmente andremo a scrivere il programma vero e proprio (vedi listato 5).
Ora non resta che programmare il Pic, scolleghiamolo dal compilatore, inseriamolo nello zoccolo professionale per demoboard nella Shield, per non danneggiare i pin del Pic ed alimentiamo BUGGY, attendiamo che svolga tutte le funzioni.

Conclusioni

In conclusione, dopo aver letto questo articolo, credo che sia abbastanza intuitivo come la MikroElektronika abbia creato una piattaforma potente e versatile, mettendo a disposizione una vasta gamma di prodotti e consigliando, per chi volesse cogliere la sfida, di ordinare non solo la BUGGY ma anche alcune Click come: Scheda Mikromedia per PIC18Fj, GSM Click, Bluetooth Click, Bluetooth Click, microSD click, MOTION Click, Compass Click, IR Click. Il tutto per realizzare prototipi completi ed affidabili. Per qualsiasi esigenza o curiosità sul progetto e sull'utilizzo del dispositivo potete scrivere all'indirizzo htsdpm@yahoo.it

Chiave d’accesso con transponder

di Giuseppe La Rosa

Il lettore di Transponder proposto in questo articolo è dotato di un’uscita a relè che si attiva al riconoscimento di uno dei 50 Trasponder RFID memorizzabili in memoria, permette di comandare elettroserrature, centraline apri cancello o centraline antifurto, grazie ad un’uscita monostabile che si attiva per circa 1 secondo, un buzzer e un LED segnalano il passaggio del Trasponder sul lettore e i vari stati della chiave.

La chiave d’accesso con Trasponder (vedi figura 1) permette l'automazione di elettroserrature e carichi elettrici di diversa natura. L'utilizzo dei Trasponder o Tag permette di ottenere un alto livello di sicurezza e la possibilità di separare la logica di controllo dal lettore di Trasponder permette di installare le due parti in zone diverse, incrementando la sicurezza anche in caso di manomissione dell'unita esterna (lettore di Trasponder); la comunicazione tra i due moduli è affidata ad una trecciola colorata a 12 fili. Il LED posto nella parte anteriore notifica all’utente l'attivazione del relè. Un buzzer avvisa l'utente tramite un suono che il passaggio del Trasponder è avvenuto correttamente. L’uso di questo progetto permette di aggiungere un Trasponder in qualsiasi momento, tramite l’uso di un Trasponder denominato CARD MASTER nei prossimi paragrafi analizzeremo la realizzazione e l’uso.

Figura 1: Foto della scheda ultimata.

La tecnologia RFID e lo standard MIFARE

Un sistema RFID è costituito da un ricetrasmettitore (lettore) e da uno o più Trasponder capaci di comunicare tra loro mediante un segnale modulato a radio frequenza (figura 2). Nel caso si utilizzi un transponder di tipo passivo (standard MIFARE) come quello da noi utilizzato in questo progetto (che non richiede a bordo una batteria per l’alimentazione), l’energia necessaria al suo funzionamento viene fornita nella fase iniziale dal ricetrasmettitore, che attraverso la sua antenna genera un opportuno campo magnetico a RF. Il Trasponder capta la RF con la propria antenna andando a caricare un piccolo condensatore che vi si trova integrato. Quando la tensione ai capi del condensatore ha superato un certo valore, il Trasponder invia al ricetrasmettitore i dati contenuti nella sua memoria modulando il segnale RF. Questi dati rappresentano, nel caso di un Trasponder a sola lettura, un codice unico, scelto tra parecchi miliardi di combinazioni possibili, che viene memorizzato sul chip durante la produzione. I dati possono essere inviati in entrambe le direzioni a seconda delle caratteristiche del transponder (ed ovviamente, del ricevitore) che può incorporare sia un chip con memoria ROM (ed in questo caso può essere solo letto) oppure una memoria di tipo EEPROM che ne consente sia la lettura che la scrittura. È facile a questo punto immaginare l’effettiva efficacia di un sistema di identificazione di questo tipo e la versatilità di un sistema di

Figura 2: Schema a blocchi sistema RFID.

raccolta dati che lo utilizzi. I sistemi RFID basati su transponder passivi hanno, ovviamente, dei vincoli; per esempio la distanza di lettura risulta piuttosto limitata a causa dei deboli segnali che vengono trasferiti; essa può variare di alcuni centimetri fino a circa 1 metro a seconda delle caratteristiche del lettore, del Trasponder ed in particolare delle dimensioni delle antenne. Con il modulo RFID-RC522 (figura 3) le distanze massime di lettura sono di circa 5 cm. I Trasponder usati in questo progetto sono due

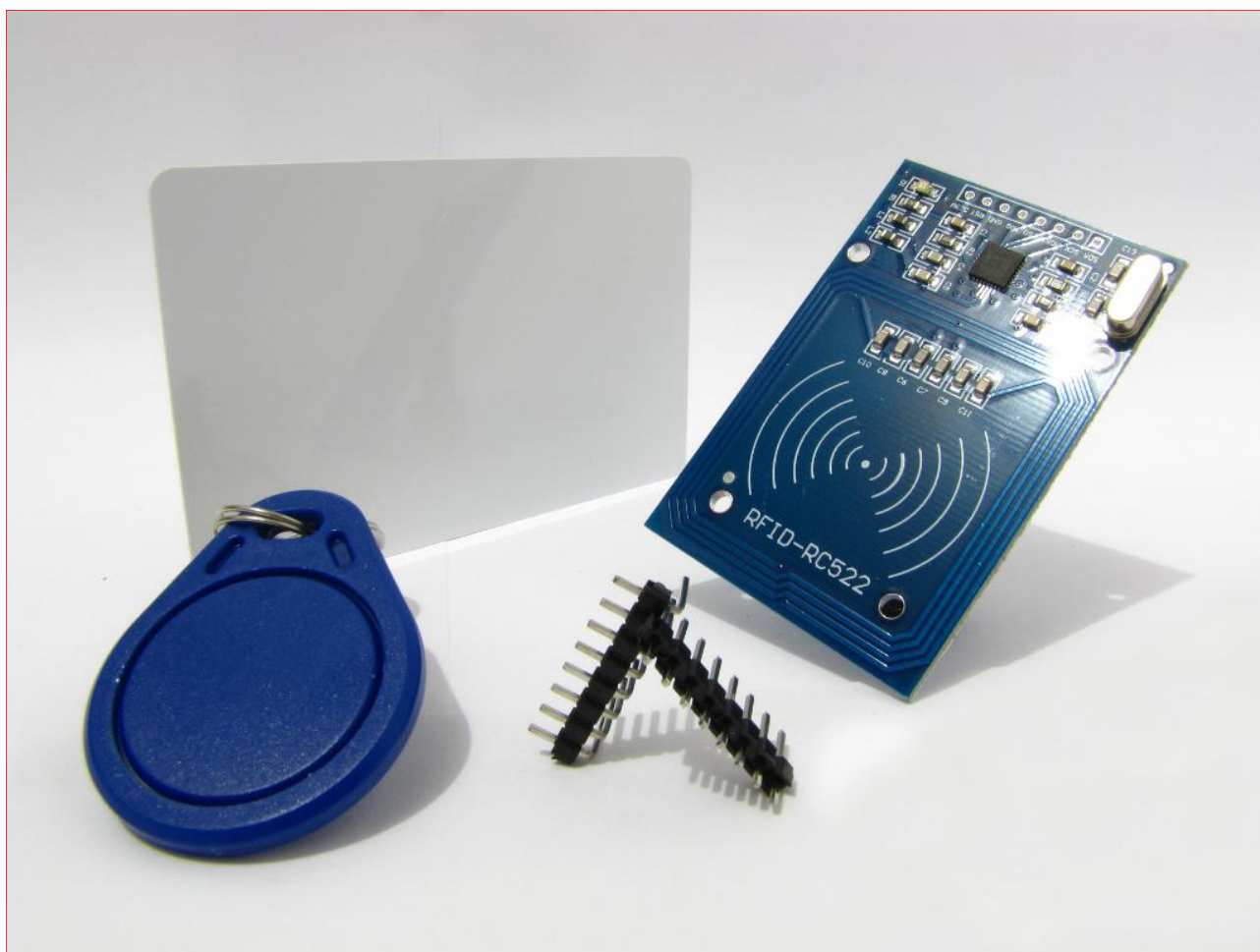



Figura 3: Modulo RFID-RC522 Trasponder passivi Card e porta chiavi.

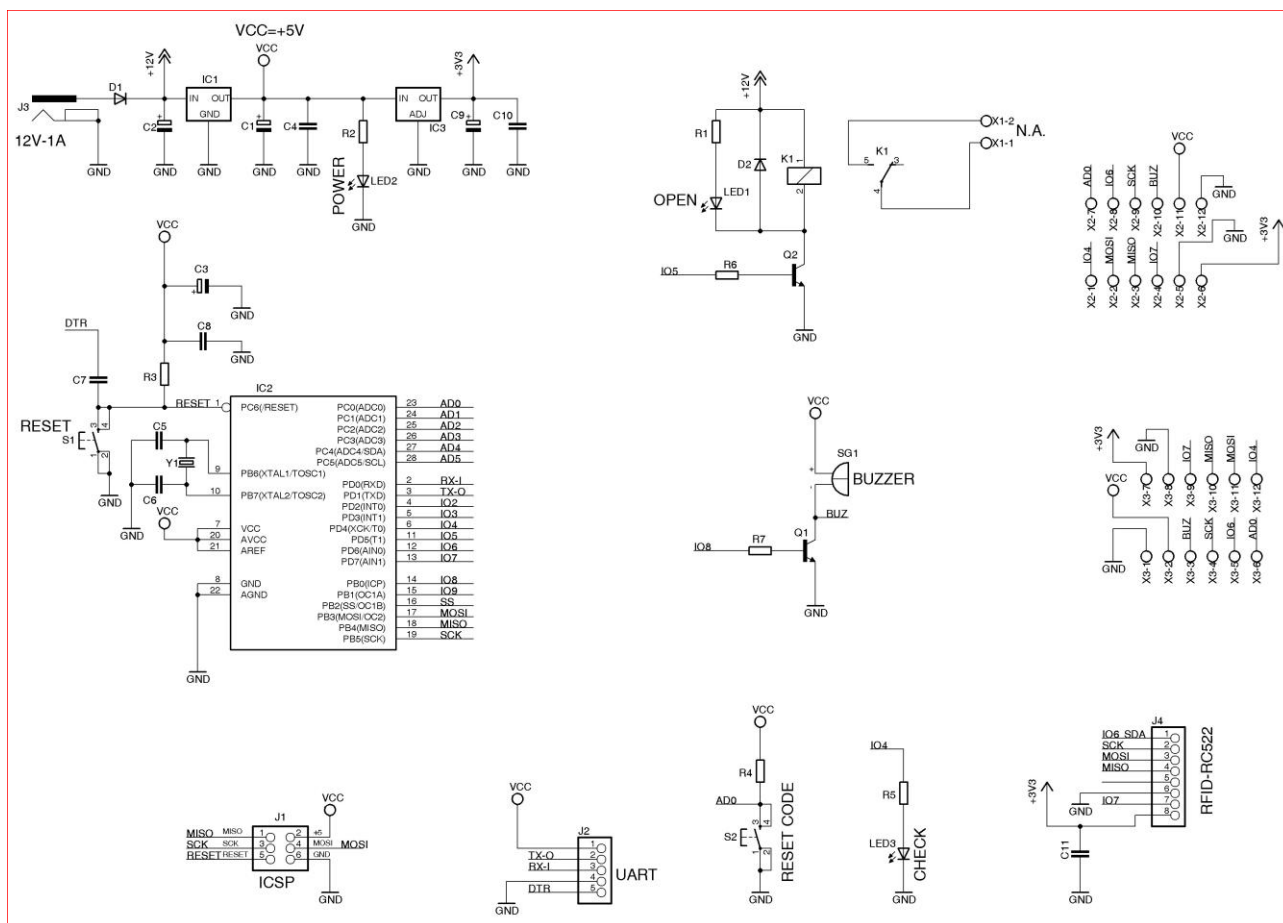



Figura 4: Schema elettrico della chiave d’accesso con Trasponder.

la Card formato 5,5 per 8,5 cm (figura 3) e il Trasponder a porta chiavi, essi supportano lo standard MIFARE. La tecnologia MIFARE, brevettata da NXP Semiconductors, è basata sullo standard ISO 14443, tipo A (RFID passivo a 13,56 MHz). Essi sono Trasponder fondamentalmente di sola memoria, dove la stessa memoria è divisa in segmenti e blocchi in cui è possibile accedere alla memoria con semplici algoritmi di controllo. Esse sono basate su Application specific integrated circuit, ed hanno quindi limitato potere computazionale. Grazie al basso costo ed all'affidabilità, questi Trasponder sono largamente usati per borsellini elettronici, controllo degli accessi, trasporti o biglietti per manifestazioni sportive.

Il circuito elettrico

In figura 4 è disegnato lo schema elettrico della chiave d’accesso con transponder. Il relè K1, il modulo RFID-RC522 (J4), il pulsante (S2), il buzzer (SG1) e il LED3 vengono gestiti da Arduino UNO, un ATMEGA328P (IC2), ed il circuito che ne risulta è molto semplice, come si evince dalla figura 4. Il relè K1 ha il compito di dare un impulso di 1 secondo all’elettroserratura connessa al morsetto X1. Poiché il relè K1 non può essere connesso direttamente alla porta del microcontrollore IC2, perché ha un assorbimento elevato, maggiore della corrente massima erogabile dal microcontrollore, per ovviare a questo problema è stato usato il transistor Q2 in configurazione d’interruttore elettronico con il relativo diodo Damper D2 che agisce da soppressore di sovratensioni, impedendo alle extra tensioni generate dalla bobina del relè di attraversare il transistor Q2. Il pulsante S2 connesso all’ingresso AD0 di IC2 ha la funzione di resettare il codice di tutti i Trasponder memorizzati nella memoria EEPROM del microcontrollore IC2. Il diodo LED (LED3) pilotato dal piedino IO4 del microcontrollore IC2 ha il compito di visualizzare tutti gli stati assunti dalla scheda insieme al buzzer SG1. Lo stabilizzatore IC1 provvede a fornire la tensione di alimentazione a 5 V

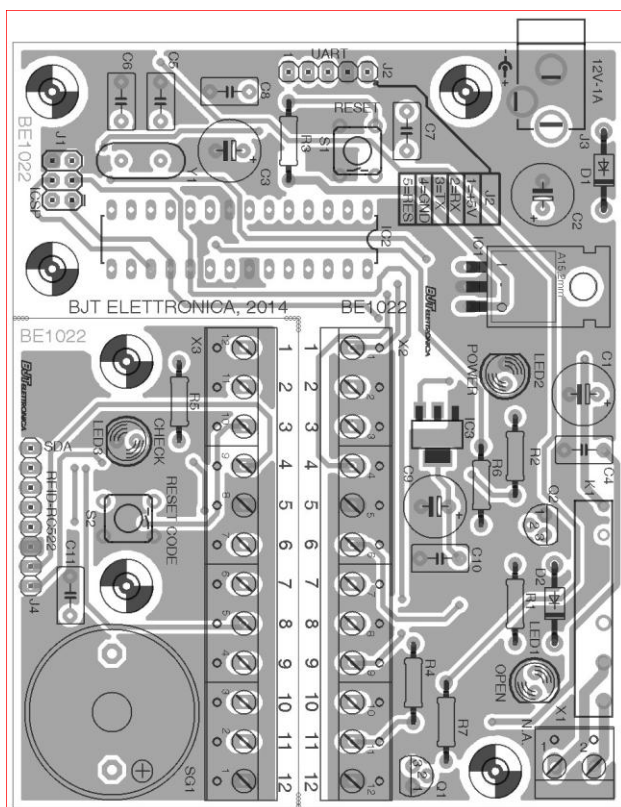



Figura 5: Disposizione dei componenti sulla piastra.

per alimentare il microcontrollore IC2, mentre lo stabilizzatore IC3 fornisce la tensione a 3,3 V che serve al funzionamento del modulo RFID-RC522 (J4). La scheda deve essere alimentata con un alimentatore stabilizzato da 12 V in grado di erogare una corrente di 1 A. Il connettore J1 serve per la programmazione In Circuit ed è utile per caricare il Bootloader di Arduino UNO.
Mentre il connettore J2, permette di caricare il firmware dall'IDE di Arduino UNO tramite convertitore seriale USB. Nel prossimo paragrafo analizzeremo il funzionamento dettagliato della scheda.

Firmware e funzionamento

Il firmware dell'ATMEGA328P è stato scritto in linguaggio C (Arduino UNO). Il sorgente del firmware è disponibile per il download sul sito della rivista ed è possibile eseguire tutte le modifiche che si vogliono. In questa applicazione è stata usata l'EEPROM interna del microcontrollore per memorizzare i 50 codici dei Trasponder o Tag. Il funzionamento è molto semplice impiegando una CARD denominata "MASTER" che viene usata per la registrazione in EEPROM di CARD nuove (fino a 50 CARD nuove) il cui codice è stato memorizzato alla variabile "CardMaster" nel sorgente e poi caricato sul microcontrollore. È possibile

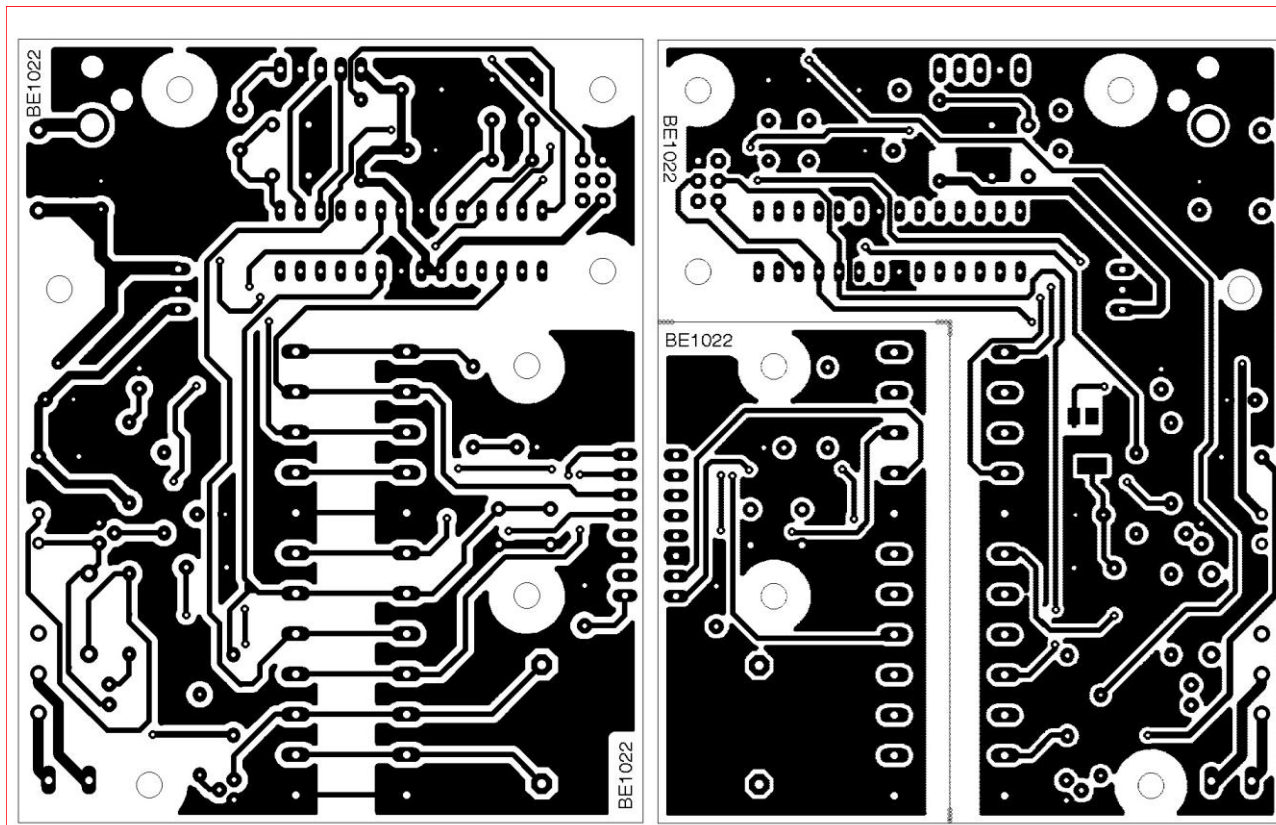



Figura 6: A destra master lato componenti a sinistra master lato saldature.

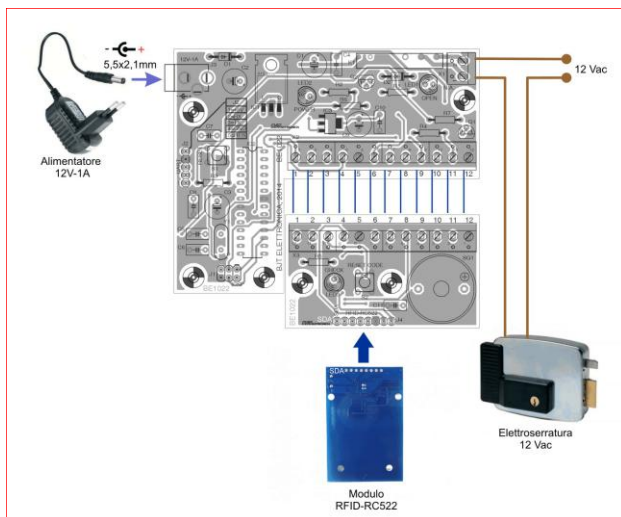



Figura 7: Schema di collegamento della chiave d'accesso con Trasponder.

registrare una CARD NUOVA basta fare rilevare alla scheda la CARD MASTER ed il LED3 comincerà a lampeggiare il Buzzer SG1 produrrà una serie di beep veloci quindi entro 6 secondi fare rilevare la NUOVA CARD e infine facendo rilevare ancora la CARD MASTER. Per leggere il codice di una CARD per poi usarla come CARD MASTER bisogna caricare il Firmware cosi com'è scaricato dal sito della rivista, connettere un convertitore seriale USB/TTL al connettore J2 aprire l'IDE di Arduino settare la porta del convertitore seriale USB/TTL, aprire il Serial Monitor (figura 9) passare la CARD sul modulo RFID-RC522 e ne segue che il codice della CARD verrà scritto sul Serial Monitor, successivamente il codice della CARD va inserito nel sorgente alla variabile "CardMaster" e in seguito il Firmware va ricaricato con il nuovo codice sulla scheda, cosi facendo abbiamo realizzato la nostra CARD MASTER utilizzabile per registrare le CARD utente. Per cancellare tutti i codici delle CARD utenti basta tenere premuto il pulsante S2 per 10 secondi e verrà eseguita la cancellazione totale dei codici delle CARD utente, questa operazione non influisce sul codice della CARD MASTER il cui codice viene conservato in memoria. Data la semplicità e i molteplici commenti inseriti nel sorgente del firmware non ci dilunghiamo nella sua descrizione.

Realizzazione della scheda e collaudo

Passiamo adesso alla costruzione della scheda che si presenta abbastanza semplice, la basetta è del tipo doppia faccia con fori metallizzati e si prepara a partire dalle tracce di figura 6.
Ottenuto il circuito stampato, iniziate a montare la scheda (seguendo il piano di montaggio figura 5) e i componenti richiesti dall'"Elenco componenti". Inserite le resistenze, in seguito i diodi D1 e D2, dopo lo stabilizzatore IC3 in tecnologia SMD. Saldate lo zoccolo per l'integrato IC2, proseguite con i condensatori non polarizzati e poi gli elettrolitici, in seguito gli STRIP J2 e J1, i pulsanti S1 e S2, i LED, e i morsetti X1, X2, X3, e il connettore J3 e per ultimo il relè K1. Il modulo RFID-RC522 va installato dal lato saldature come visibile in figura 8 utilizzando uno STRIP femmina, questo permette un facile alloggiamento a pannello su qualsiasi contenitore. Se si dispone di un ATMEGA328P già con Bootloader caricato si può passare al caricamento del firmware tramite un convertitore seriale USB/TTL e collegarlo al connettore J2 presente sulla scheda. Altrimenti prima bisogna caricare tramite un programmatore il Bootloader dal connettore J1. Eseguite il collegamento com'è rappresentato in figura 7, collegate

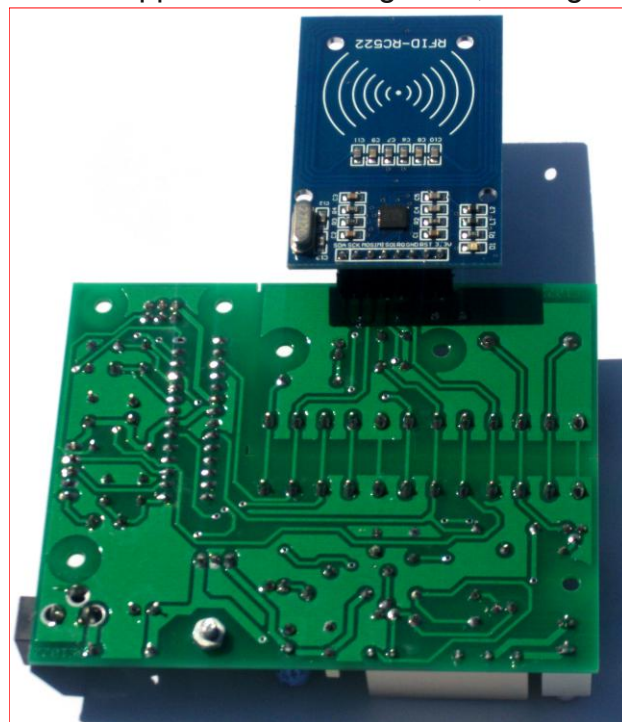



Figura 8: Scheda vista da lato saldature.

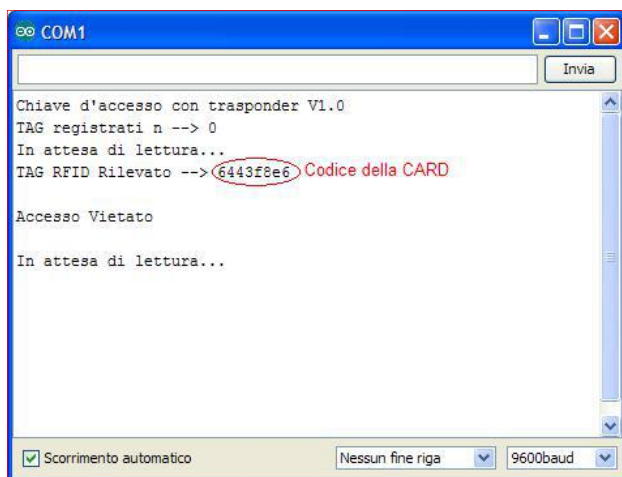



Figura 9: Schermata del Serial Monitor dell'IDE Arduino.

un alimentatore a 12 V in grado di erogare 1 A. A questo punto potete collaudare la scheda, basta passare una CARD sul modulo RFID-RC522 se il circuito funziona, il LED3 si illuminerà e il buzzer SG1 produrrà beep lungo ad indicare che la CARD non è registrata in memoria, effettuata questa operazione, bisogna creare la CARD MASTER come descritto nel paragrafo precedente. Successivamente programmate tutte le CARD che vi occorrono come descritto nel paragrafo "Firmware e funzionamento". Se il collaudo ha dato esito positivo la parte del circuito stampato contenente il pulsante, il modulo RFID-RC522 e il buzzer, può essere tagliata e staccata dalla scheda per essere fissata in un contenitore assieme al modulo RFID-RC522 (vedi figura 7) in modo da tenerla divisa dalla scheda logica. Questo permette d'installare il modulo RFID-RC522 accanto alla porta o cancello e la scheda logica in altro luogo meno accessibile in modo da non essere manomessa facilmente.

Conclusione

Tutti i file per la realizzazione, come già è stato detto, sono disponibili sul sito di Fare Elettronica. Come avete letto in precedenza le applicazioni di questo progetto sono innumerevoli e spaziano dal controllo accessi ai sistemi anti intrusione, pertanto il progetto è ampiamente adeguabile alle vostre esigenze. Il sorgente del firmware è scritto in linguaggio C per Arduino UNO ed è facilmente modificabile e reimpiegabile per le proprie esigenze. Scaricare i file utili alla realizzazione clicca qui:

http://larosagiuseppe.altervista.org/BE1022.rar
Guardare il video della scheda in funzione clicca qui:
https://www.youtube.com/watch?v=rml5qc-wdslg&feature=youtu.be

Elenco componenti

R1	1 KΩ 1/4 W
R2	470 Ω 1/4 W
R3÷R4	10 KΩ 1/4 W
R5	330 Ω 1/4 W
R6	15 KΩ 1/4 W
R7	12 KΩ 1/4 W
C1÷C3	100 µF 35 V elettrol.
C4	100 nF poliestere
C5÷C6	22 pF ceramico
C7÷C8	100 nF poliestere
C9	100 µF 35 V elettrol.
C10÷C11	100 nF poliestere
D1÷D2	1N4007 diodo
Q1÷Q2	BC337
IC1	L7805CV
IC2	ATMEGA328P
IC3	LD117AS33TR SMD
LED1	Led 5 mm giallo
LED2	Led 5 mm verde
LED3	Led 5 mm rosso
Y1	Quarzo 16 MHz
K1	Relè 12 V/6 A
S1÷S2	Pulsante C.S.
SG1	Buzzer passo 15 mm
J1	Strip maschio 3+3 pin
J2	Strip maschio 5 pin
J3	Presa DC 90° 5,5x2,1mm
J4	Strip femmina 8 pin
X1	Morsetto 2 poli
X2÷X3	Morsetto 12 poli
N.1	Zoccolo 14+14 pin
N.1	Vite 3x15 mm più dado
N.1	Modulo RFID-RC522

Listato 1

```
/*
Chiave d'accesso con transponder
By LA ROSA GIUSEPPE
07/10/2014
 --------------------------------------------------
 * Layout dei pin è il seguente:
 * Segnali    Pin                   Pin
 *            Arduino Uno           MFRC522 board
 * ------------------------------------------------
 * Reset      7                     RST
 * SPI SS     6                     SDA
 * SPI MOSI   11                    MOSI
 * SPI MISO   12                    MISO
 * SPI SCK    13                    SCK
 *
 * -----------------------------------------------------------------------------------------------------------------------------
 * Funzionamento con l'uso di card denominata "MASTER" per la registrazione in EEPROM di carte nuove (ora impostato fino 50)
 * Per semplicità il codice di questa carta verrà inserita nella variabile CardMaster="97cd83bf"
 * Con la CARD MASTER si possono abilitare nuove card.
 * è stata aggiunta una uscita su cui collegato un led LED3 ed una uscita dove collegato un buzzer SG1 per capire le sequenze
 * senza utilizzare la seriale.
 * Per registrare una CARD : rilevare la CARD MASTER quindi entro 6 secondi rilevare la NUOVA CARD e terminare ancora con la CARD
MASTER
 * per reset generale      : tenere premuto il pulsante S2 10 sec. Il reset non influisce sul codice della CARD MASTER
 * ------------------------------------------------------------------------------------------------------------------------------
 */
#include <SPI.h>
#include <MFRC522.h>
#include <EEPROM.h>
#define SS_PIN 6
#define RST_PIN 7
#define attiva_rele 5          // pin relè
#define attiva_temporele 1500  // tempo in millisecondi attuazione rele
#define attiva_led 4           // pin led
#define reset_gen A0           // pulsante da premere per 10 sec per azzerare i codici in memoria
#define buzzer 8               // pin buzzer
MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance.

// ---------------------------------------------------------------  MASTER
String CardMaster="97cd83bf"; // INSERIRE QUI IL CODICE DELLA CARTA MASTER
// ----------------------------------------------------------------------
long previousMillis = 0;
long previousMillis1 = 0;
long interval = 10;     // intervallo lettura RFID
long interval1 = 10000; // intervallo oltre al quale si esce dalla programmazione (modificabile in mills)
unsigned long tempo_rst = 800000; // intervallo prima reset generale (modificabile)
unsigned long cnt=0;
int  Programmazione=0;
byte pch = 0; // puntatore card registrate
String uid_s1;

String uidRFID[50];
```

```
void setup() {
     Serial.begin(9600); // Inizializza la comunicazione seriale con il PC
    SPI.begin();        // Init SPI bus
    mfrc522.PCD_Init(); // Init MFRC522 card
    pinMode(attiva_rele, OUTPUT);
    pinMode(attiva_led, OUTPUT);
    pinMode(reset_gen,INPUT);
    pinMode(buzzer,OUTPUT);
 // lettura da EEPROM -----------------------------------------------------------------------------------------------------------
 Serial.println("Chiave d'accesso con trasponder V1.0");
  pch=EEPROM.read(0);

  if (pch == 123) {    // leggo valori
     pch=EEPROM.read(1);
  Serial.print("TAG registrati n --> ");
   Serial.println(pch);

     int i=1;
     while(i<=pch) {
       uidRFID[i-1]="";
       for (int p=i*8-8+2;p<i*8+2;p++) {
         uidRFID[i-1]=uidRFID[i-1]+char(EEPROM.read(p));
         }
      Serial.println(uidRFID[i-1]);
      i++;
      }
  }
  else {                        // mai scritta EEPROM
     EEPROM.write (0,123);
     EEPROM.write (1,0);
  }

 // Fine lettura da EEPROM -----------------------------------------------------------------------------------------------
    Serial.println("In attesa di lettura...");
}
void loop() {
  // scansione lettore ogni "interval" in mS ---------------------
 unsigned long currentMillis = millis();
 // se premuto reset ...
 while (!digitalRead(reset_gen)) {
     // conteggio 10 sec prima di resettare
    while (!digitalRead(reset_gen)) {
              cnt++;
              if (cnt>tempo_rst) {
                 // reset confermato
                 digitalWrite(attiva_led,HIGH);
                 digitalWrite(buzzer,HIGH);
                 pch=0;
                 EEPROM.write (1,0);
                 cnt=0;
                    for (int q=0;q<8;q++) {
                         digitalWrite(attiva_led,HIGH);
                         digitalWrite(buzzer,HIGH);
```

```
                        delay(100);
                        digitalWrite(attiva_led,LOW);
                        digitalWrite(buzzer,LOW);
                        delay(100);
                  }
             }
             break;
   }
}
cnt=0;

if(Programmazione > 0 && currentMillis - previousMillis1 > interval1) {
   previousMillis1 = currentMillis;
   Programmazione = 0;
   digitalWrite(attiva_led,LOW);
   digitalWrite(buzzer,HIGH);
   delay(2000);
   digitalWrite(buzzer,LOW);
   Serial.println("Programmazione Cancellata");
}
if(currentMillis - previousMillis > interval) {
   previousMillis = currentMillis;

   // i due if seguenti interrompono l'esecuzione se le condizioni di presenza card non è verificata velocizzando l'esecuzione
   // Look for new cards
   if ( ! mfrc522.PICC_IsNewCardPresent()) {
       return;
   }
    // Select one of the cards
   if ( ! mfrc522.PICC_ReadCardSerial()) {
       return;
   }

  // lettura codice card ----------------------------------------
  String uid_s = "";

  if (!mfrc522.PICC_IsNewCardPresent() && !mfrc522.PICC_ReadCardSerial()) {

      for (byte i = 0; i < mfrc522.uid.size; i++) {

         String uid_a = String(mfrc522.uid.uidByte[i] < 0x10 ? "0" : "");
         String uid_b = String(mfrc522.uid.uidByte[i], HEX);

         uid_s = uid_s+uid_a+uid_b;
     }
  }
  // in uid_s la stringa con il codice della card letta ---------

  Serial.print("TAG RFID Rilevato --> ");
  Serial.println(uid_s);
  Serial.println("");
  boolean controllo = false;

// ---------------------------------------------------- rilevazione codici
```

```
if (uid_s==CardMaster) {                      // entra in programmazione
    digitalWrite(attiva_led, HIGH);
    delay(100);
    digitalWrite(attiva_led, LOW);
    previousMillis1 = millis();
    switch(Programmazione){
       case 0:
           Programmazione=1;
           Serial.println("In Programmazione......");
           for (int q=0;q<8;q++) {
                 digitalWrite(attiva_led,HIGH);
                 digitalWrite(buzzer,HIGH);
                 delay(100);
                 digitalWrite(attiva_led,LOW);
                 digitalWrite(buzzer,LOW);
                 delay(100);
           }
           digitalWrite(attiva_led,HIGH);

       break;
       case 2:
           // Programmazione nuovo codice confermato
           // in uid_s1 il valore da scrivere
           uidRFID[pch]=uid_s1;
           // scrivi in EEPROM
           EEPROM.write (1,pch+1);
           for (int p=0;p<8;p++) {
             EEPROM.write(pch*8+2+p,byte(uid_s1.charAt(p)));
           }
           pch++;
           Programmazione=0;
           Serial.println("Codice Memorizzato");
           digitalWrite(buzzer,HIGH);
           delay(100);
           digitalWrite(buzzer,LOW);
           digitalWrite(attiva_led,LOW);
           delay(1000);
           break;
       case 3:
           // ...
           break;

        default: break;
    }
}

else {                                        // valuta codici in memoria
   for (int i = 0; i < pch; i++){

     if(uidRFID[i] == uid_s){

          Serial.println("Accesso Consentito");
          openDoor();
          controllo = true;
```

```
            break;
         }
       }

      if(!controllo){
         if (Programmazione==1) {
             for (int q=0;q<3;q++) {
                   digitalWrite(attiva_led,LOW);
                   digitalWrite(buzzer,HIGH);
                   delay(100);
                   digitalWrite(attiva_led,HIGH);
                   digitalWrite(buzzer,LOW);
                   delay(100);
             }
            Serial.println("Card da memorizzare -> Confermare con CARD MASTER");
            uid_s1 = uid_s;
            Programmazione=2;
            delay(500);
         }
         else {
            Serial.println("Accesso Vietato");
            digitalWrite(buzzer,HIGH);
            digitalWrite(attiva_led,HIGH);
            delay(2000);
            digitalWrite(buzzer,LOW);
            digitalWrite(attiva_led,LOW);
         }
      }
  }
   Serial.println();
   Serial.println("In attesa di lettura...");

 } // chiude attesa mills
}  // chiude loop

// ---------------------------------------- Funzioni Rele'

void openDoor(){

  digitalWrite(attiva_rele, HIGH);
  digitalWrite(attiva_led, HIGH);
   Serial.println("Rele' ON");
  digitalWrite(buzzer,HIGH);
  delay(125);
  digitalWrite(buzzer,LOW);
  delay(attiva_temporele);
  digitalWrite(attiva_rele, LOW);
  digitalWrite(attiva_led, LOW);
  Serial.println("Rele' OFF");
  delay(3500);
}
```

L'OROLOGIO PER VINIFICAZIONE

di Girolamo D'Orio

Realizzazione di un dispositivo per la visualizzazione di ora-data-temperatura-cronometro-cronometro alla rovescia. Visualizzazione su 6 display 7 segmenti tramite rotary encoder, selezione di 6 menù per regolazioni e funzioni. Menù indicati da rispettivo led.

Questa realizzazione nasce per aiutare il cantiniere durante le varie fasi di lavorazione del vino. L'obbiettivo prefissato in fase progettuale è di rendere ben visibile la visualizzazione dei display anche da lontano, il tutto deve essere racchiuso in un pcb di misura 16x10cm e di realizzare, soprattutto, con quello che è disponibile in cassettiera, dato che ho deciso di costruirlo durante il periodo natalizio. Non si tratta del solito orologio-datario esso, infatti, comprende anche un cronometro e un cronometro alla rovescia. Questo perché nel mio lavoro è fondamentale una misura del tempo per compiere alcune lavorazioni al vino. Per esempio in fase di fermentazione occorre "rimontare" il vaso vinario diverse volte al giorno. Con il termine "rimontaggio" si intende prelevare il liquido dalla parte bassa del vaso vinario e pomparla alla parte alta dello stesso. Un irroratore permetterà quindi di bagnare in modo omogeneo il "cappello della vinaccia" in modo da aumentare l'estrazione dalla buccia e non farla asciugare per lungo tempo. Dato che la massa va "rimontata" al massimo per il 75% del liquido, avendo nota la massa e la portata della pompa espressa in Qt/h occorre misurare il tempo. Per esempio, io ho contenitori di 200 hl e con le mie pompe, per ogni "rimontaggio" ho calcolato che la fase dura 15 minuti. Le cisterne da rimonta-

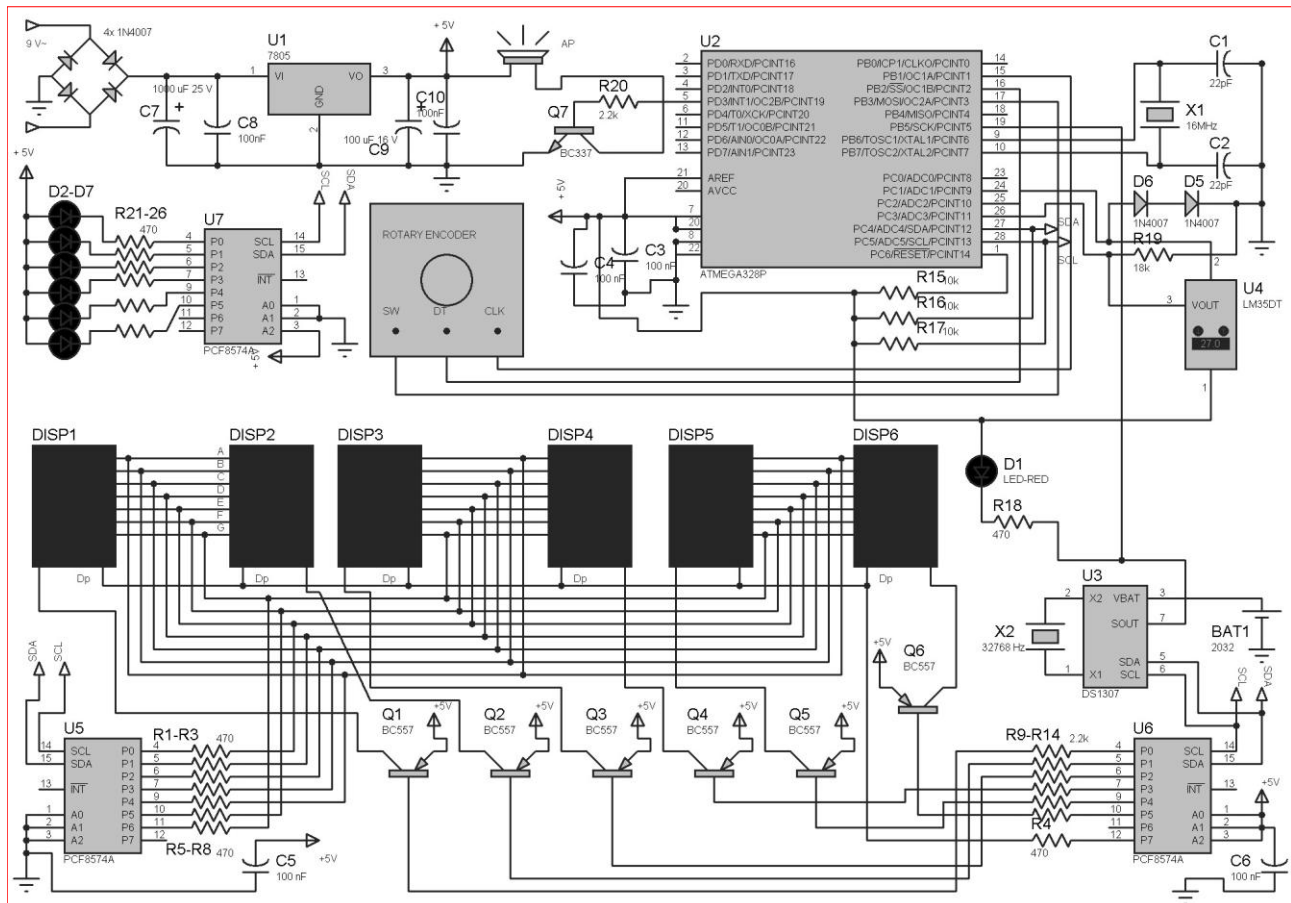



Figura 1: Schema elettrico.

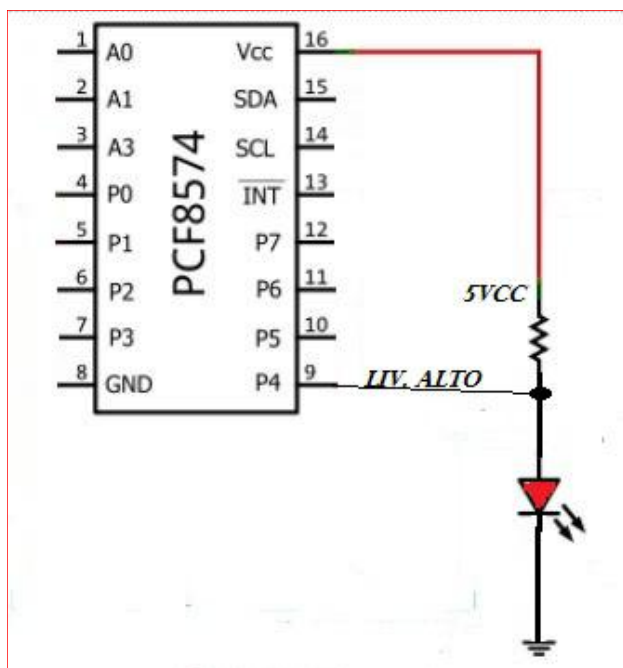



Figura 2: Esempio.

re sono molte, questo procedimento non può avvenire uguale per tutti i vasi vinari, cambia a seconda del tipo di uva che stiamo lavorando, quindi per ottenere il massimo, vengono decisi tempi diversi per ogni singolo vaso vinario. Ho sempre tenuto il tempo guardando l'orologio, ma spesso capita di avere il dubbio sull'orario di partenza… capita anche di guardare l'orologio in ritardo… Detto ciò ho ritenuto giusta la presenza della funzione cronometro alla rovescia, e ho fatto in modo che allo scadere del conteggio il cantiniere viene avvisato acusticamente con un beep molto rumoroso negli ultimi 5 secondi del conteggio. La funzione cronometro tradizionale l'ho inserita per il seguente motivo: le pompe di cantina possono essere di diversi tipi, ognuna delle quali ha le sue usure nel tempo. Quindi tenerle sempre efficienti può far risparmiare tempo a fine giornata, quando si lavora con masse di mosto in fermentazione intorno ai 2000-2500 hl come nel mio caso. Quindi il cronometro tradizionale serve per misurare quanto tempo impiega la pompa a travasare il vino da un vaso vinario all'altro. Se noto un aumento considerevole del tempo durante questa fase di lavoro andrò ad indagare se c'è un problema nella pompa. Quindi, ricapitolando, la visualizzazione necessita di display 7 segmenti il più grandi possibili e luminosi, l'orologio deve essere posizionato in un punto alto per permettere la visualizzazione da ogni posizione, i comandi devono essere a altezza uomo, pratici e manovrabili anche indossando guanti bagnati.

Descrizione dell'hardware

L'alimentatore costituito dal trasformatore, ponte a diodi, stabilizzatore a 5Vcc e contornato dalle capacità di livellamento e di filtraggio, alimenta tutto il circuito. L'assorbimento è minimo, considerando infatti che i display sono gestiti in multiplaxing, avremo al massimo un consumo di 7 led, più led lampeggio ad 1Hz e un led per il menù. Il picco massimo di assorbimento lo avremo solo quando Q7 piloterà l'altoparlante per 5 secondi. Si stà parlando di un assorbimento compreso tra 100-150 mA, quindi ho ritenuto inutile "alettare" lo stabilizzatore U1 dato che da calcolo non dovrà che dissipare centinaia di mW. Avrete

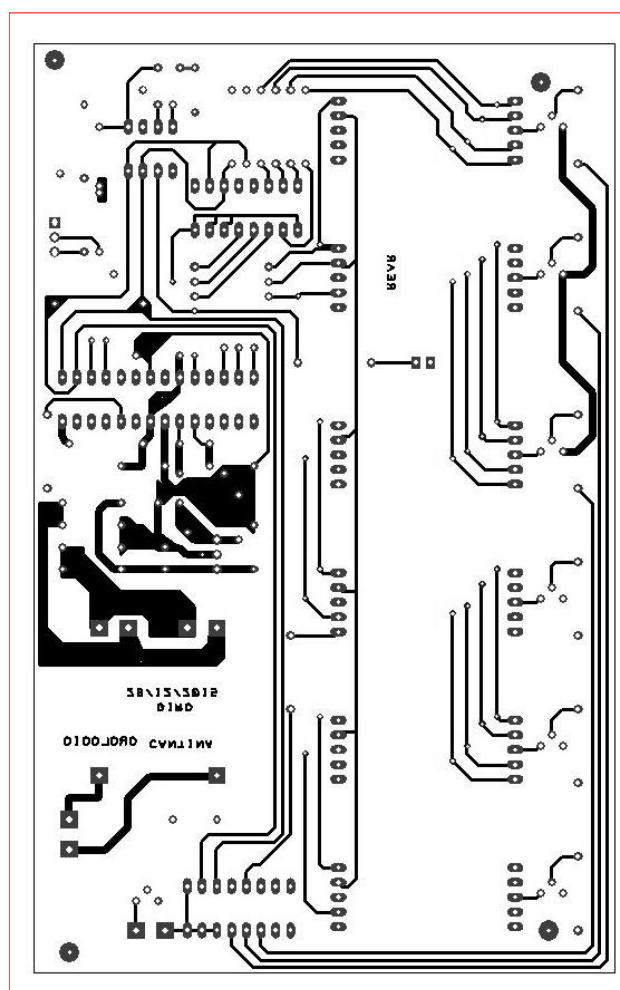



Figura 3: Master lato piste.

notato sicuramente un modo inusuale dei pilotaggio display in multiplexing: in genere si ricorre ad integrati BCD driver, per esempio il cd4511 o SN7447. Non avendoli a disposizione e volendo gestire anche i punti decimali di ogni singolo display, ho scelto il tuttofare, come lo chiamo io, il PCF8574AN. Essendo un I/O Expander con comunicazione I2C, il suo indirizzo è configurabile gestendo i suoi tre pin 1, 2, 3 con livelli logici alti e bassi. Quindi ho optato per usare U5 per gestire i singoli segmenti dei display, U6 per gestire l'anodo comune e punti decimali, infine U7 per gestire i 6 led che indicano il menù su quale andremo ad interagire. Dallo schema si nota che tali integrati sono usati non nel modo tradizionale e questo spiega il motivo perché la scelta dei display ad anodo comune. Nel metodo tradizionale, per accendere un led necessiterebbe la seguente configurazione, una resistenza di pull-up in quanto l'integrato non può supportare tali correnti. Quindi adottando la

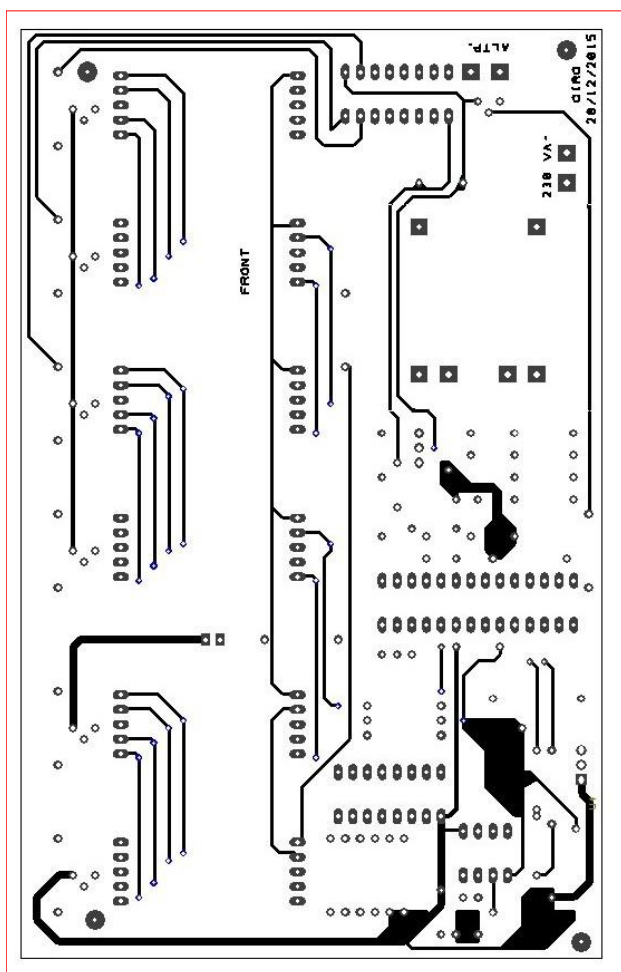



Figura 4: Master lato componenti.

Elenco componenti

Componente	Valore
R1-R8, R18,R21-R26	470 Ω ¼ W
R15-R17	10 KΩ ¼ W
R9-R14,R20	2,2 KΩ ¼ W
R18	18 KΩ ¼ W
C7	1000 uF 24V elettrolitico
C9	100 uF 16V elettrolitico
C1,C2	22 pF poliestere
C3-C6,C8, C10,C11	100 nF poliestere
U1	7805
U2	ATmega328p-pu
U3	Ds1307
U4	LM35dt
U5-U7	PCF8574AN
D1-D6	1N4007
Q1-Q6	BC557 PNP
Q7	BC337NPN
XTAL	16 Mhz
XTAL 2	32768 Hz
Display 1-6	SA10-21SRWA
Led1	Led Rosso 15mA
Led2-7	Led Verde 15ma
TR1	Trasformatore VB 3.2/2/9
Ap	Altoparlante 8 Ω
buzzer	

soluzione come da mio schema sono andato a risparmiare collegamenti e resistenze di pull-up. Sempre con comunicazione I2C avviene il dialogo con U3 che ha il compito di tenere il conteggio del tempo di data e ora. Dal suo pin 7 prelevo l'onda quadra richiamata dal microcontrollore ATmega328p-pu ad 1 Hz, in modo da pilotare direttamente LD1 e comunicare allo stesso tempo al micro il livello logico dell'onda quadra generata da ds1307. LD1 si accenderà solo quando l'onda quadra è a livello logico basso. U4, la sonda di temperatura LM35dt, in package TO-220, è configurata in Full-Range per rilevare temperature anche inferiori allo 0. Non sfrutta solo un pin ADC del micro, ma due. I

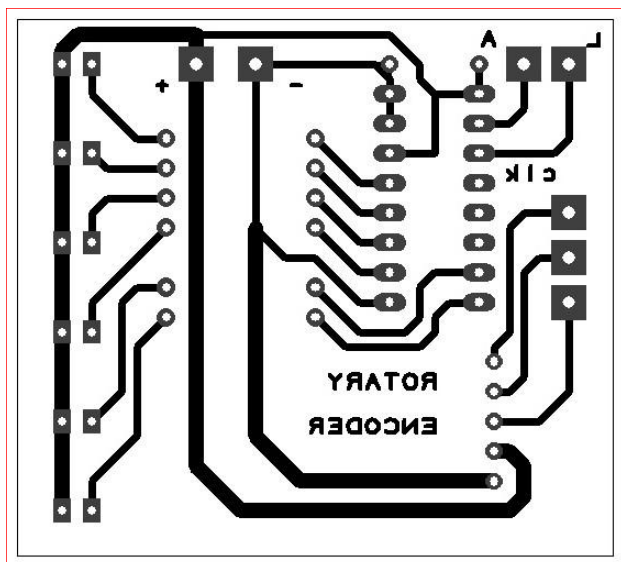



Figura 5: Master comando remoto (45x51 mm.)

due diodi D5-D6 e la resistenza R19 sono consigliate dal datasheets di tale componente per sfruttare proprio la modalità Full-Range. Il Pin PWM di U2 pilota Q7 per permettere appunto all'altoparlante di emettere proprio un bel Beep che suonerà come ho sopracitato agli ultimi 5 secondi del conteggio del cronometro alla rovescia. Inoltre suonerà per un breve istante ogni volta che premiamo il pulsante incorporato nel rotary encoder per segnalarci l'avvenuta pressione del pulsante stesso quando interagiamo con i vari menù. Per rientrare nello spazio di 16x10cm il pcb è un doppia faccia.

Versione con comandi e menu integrati nel pcb

Mia moglie, dopo aver visto il dispositivo installato in cantina, mi ha detto che gli poteva essere di aiuto in cucina, quando sfodera la sua arte culinaria… Mi ha chiesto di replicarlo ma senza comando remoto, i comandi e Led-menù devono essere integrati nel pcb. Quindi per trovare spazio ho eliminato il trasformatore dal pcb e alimenterò con un trasformatore da inserire nella presa domestica. Per chi avesse esigenze di questo tipo pubblico anche la versione con comandi integrati. Al posto dell'altoparlante, che nella versione precedente è installato esternamente al pcb, sono riuscito ad inserire un buzzer all'interno del pcb stesso e senza fare modifiche al software. Dalla cantina si passa alla cucina…

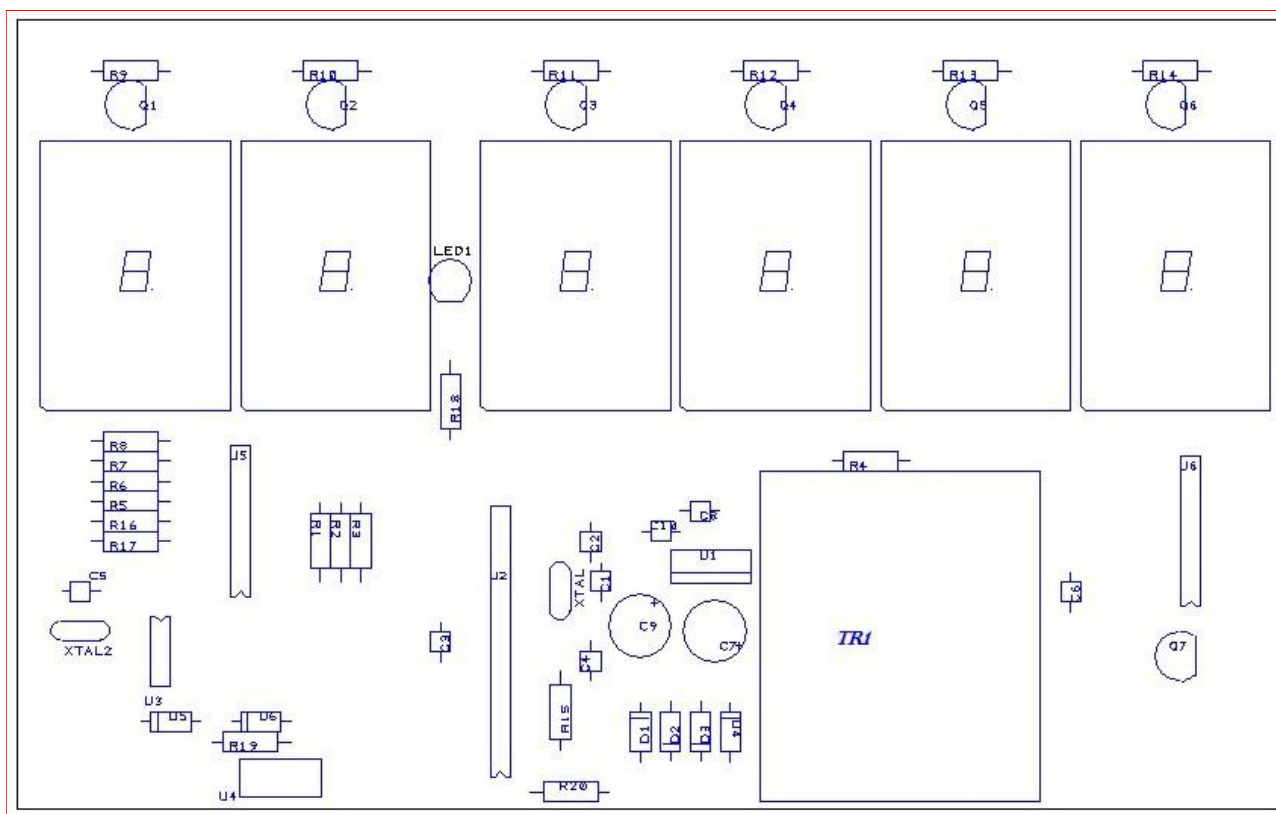



Figura 6: Montaggio.

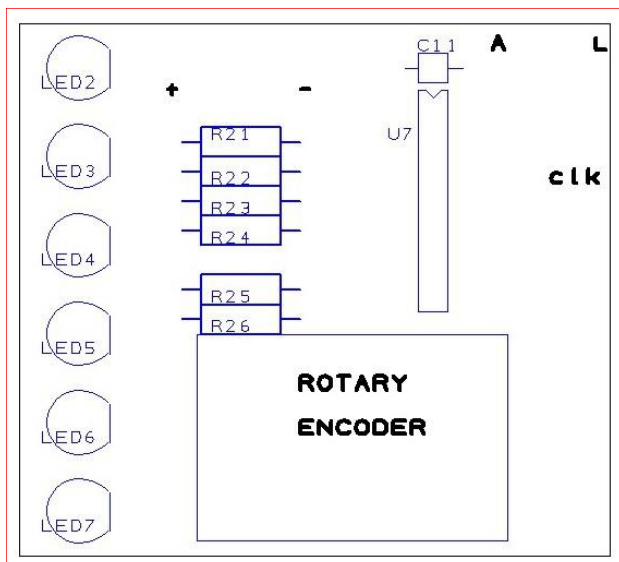



Figura 7: Montaggio comando remoto.

Descrizione del software per Arduino

Il sorgente è un po' lungo, circa 1450 righe, ma occupa solamente il 62% dello spazio disponibile. Non potendolo pubblicare inte-

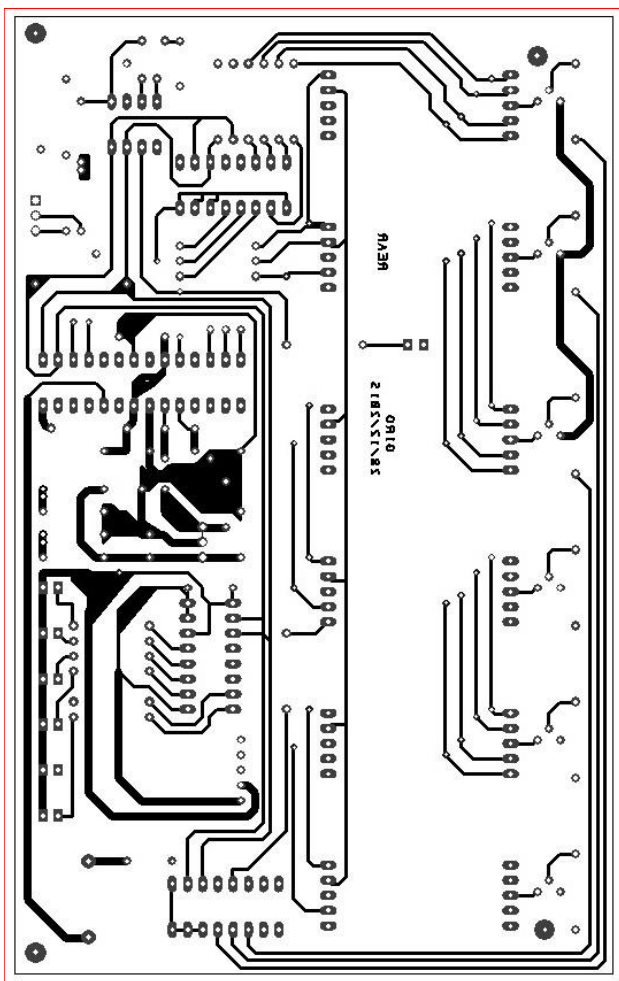



Figura 8: Master lato piste V2.

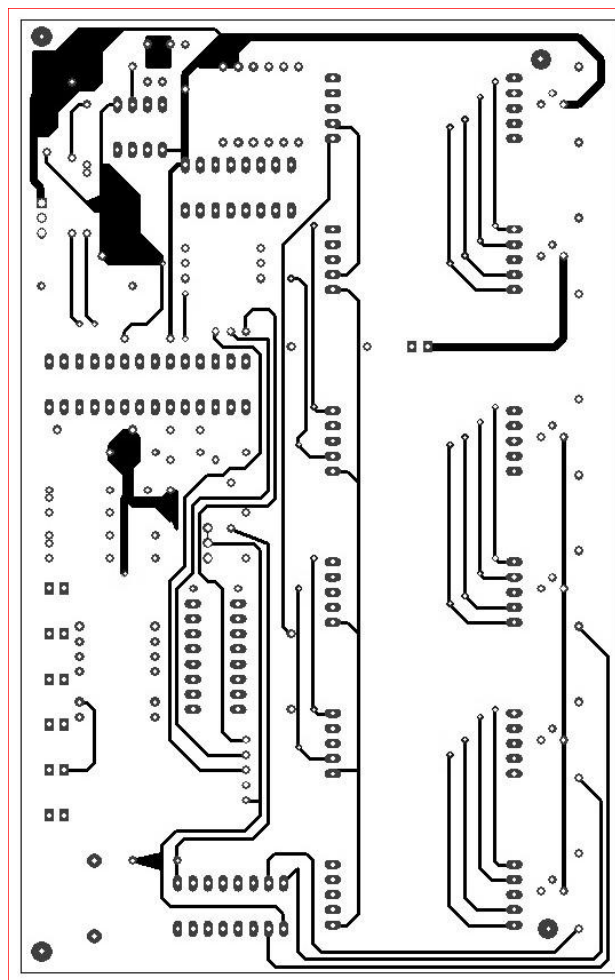



Figura 9: Master lato componenti V2.

ramente troverete il download in fondo alla descrizione. Riassumendolo brevemente ho creato 10 sub-routine che verranno richiamate all'occorrenza durante il normale svolgimento del programma. Per prima cosa ho creato i caratteri da inviare tramite I2c alll'integrato U5.

Esempio del numero 3:

Wire.beginTransmission(0x38); //indirizzo I2c di U5,avvio trasmissivo dati

Wire.write((byte)0b10100001); // invio valore in binario

Wire.endTransmission(); // fine trasmissione dati

Immaginate che la seconda riga in realtà dice ad U5 come deve impostare i livelli logici delle sue porte. Dato che sto lavorando con anodo comune, i pin con il livello logico basso permetteranno quindi di accendere il segmento corrispondente in quanto avviene una differenza di potenziale. Se ci pensate bene niente di più semplice. Avendo solo un pulsante per fare tutte le impostazioni ho

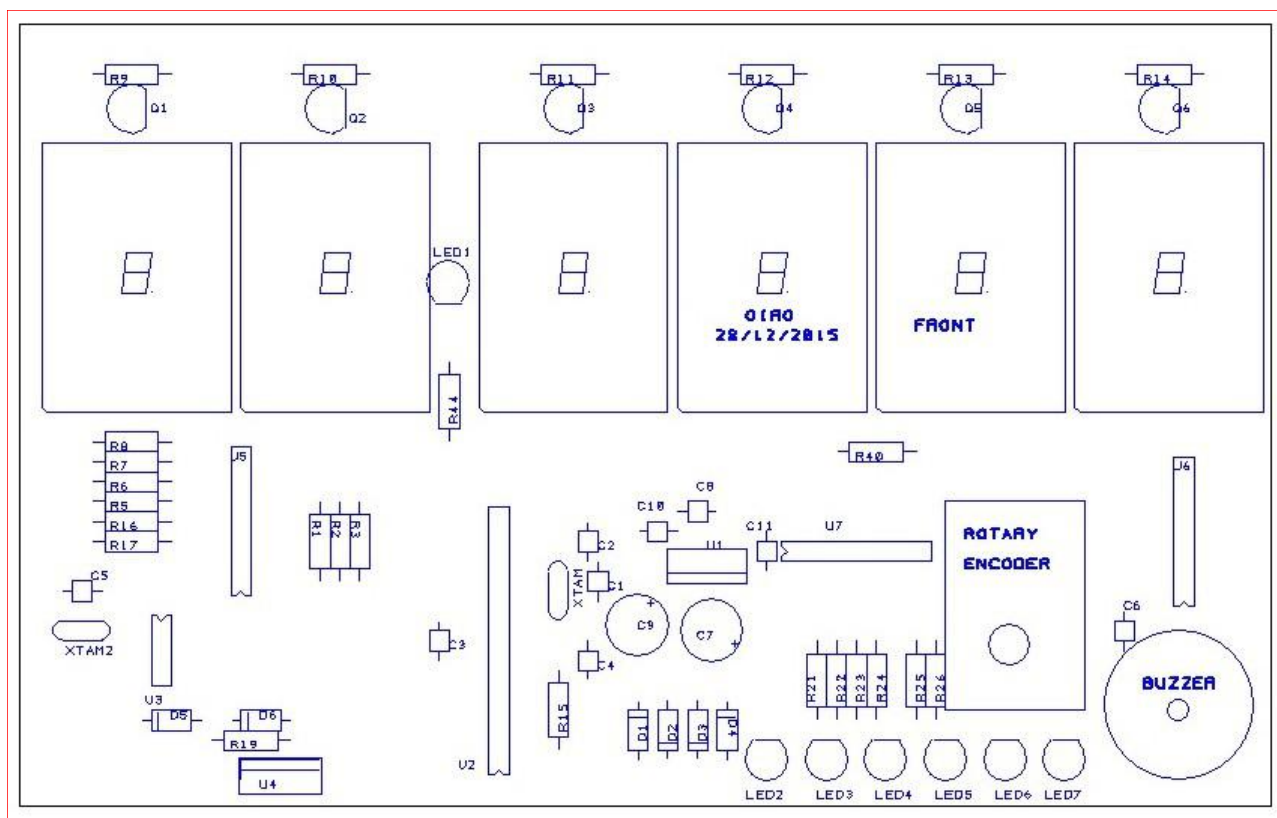



Figura 10: Montaggio V2.

dovuto fare uso di molte variabili per entrare ed uscire delle sub-routine. Il sorgente vale più di mille parole, guardiamo cosa avviene nel Void Loop, cioè cosa farà U2 nel suo ciclo infinito. Il conteggio del tempo per il cronometro alla rovescia, sfrutta un principio diverso. Il microcontrollore andrà a dialogare con U3 abilitando il suo pin 7 con un onda quadra ad 1Hz e il micro ogni volta che la vede a livello logico alto decrementa la variabile dei secondi. Con questo metodo vado a decrementare il tempo che è stato reimpostato a vostra scelta nell'apposito menù. Di default il conto alla rovescia inizia da 15 minuti.

Consigli per il montaggio

Il montaggio non presenta particolari difficoltà, consiglio l'uso di una punta sottile per saldare gli zoccoli degli integrati sul lato componenti se siete impossibilitati come me di realizzare il trattamento per rendere i fori metallizzati. Il trasformatore l'ho montato nel lato piste per avvicinare il più possibile i display al coperchio della scatola, in quanto il trasformatore è più alto dei display stessi. Sempre sul lato piste ho saldato il porta-batteria per la 2032 che permette di non perdere l'orario in caso di blackout. L'altoparlante, come il sensore di temperatura LM35dt, li ho collegati con filo per tenerli esterni al contenitore. Per collegare il comando remoto al PCB ho usato un cavo di rete e consiglio di non superare la lunghezza di 3 Mt, altrimenti il dispositivo potrebbe non funzionare correttamente. Ancora non ho realizzato fisicamente la seconda versione, ecco il Layer in 3d per farvi rendere conto come verrà.

Conclusioni

Il classico orologio Home-Made con qualcosa in più, chiaramente tutti abbiamo uno Smartphone che svolge tutte queste funzioni, ma la soddisfazione di averlo fatto da noi è un'altra cosa…

Link del video in fase di collaudo in casa:

https://www.youtube.com/watch?v=_9cZ4O6x-t4

Buon divertimento e buona realizzazione a tutti. Giro.

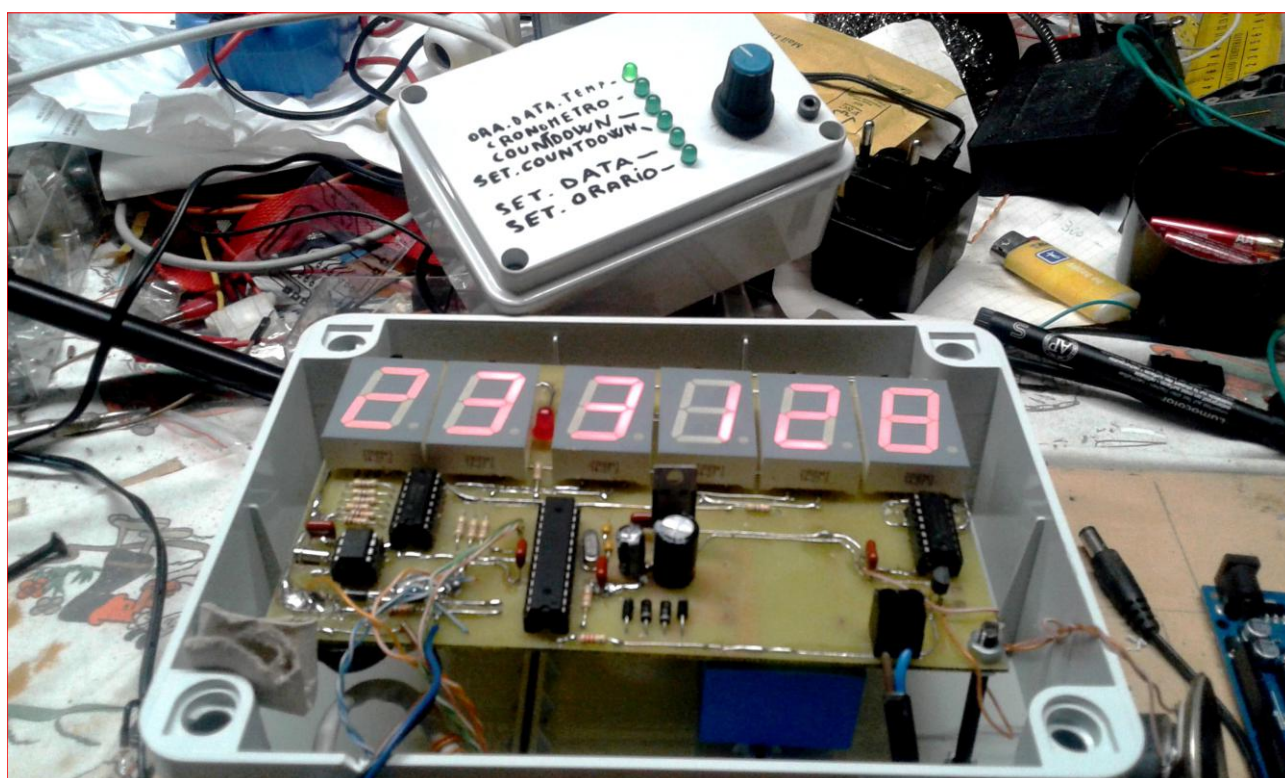



Figura 11: Realizzazione.

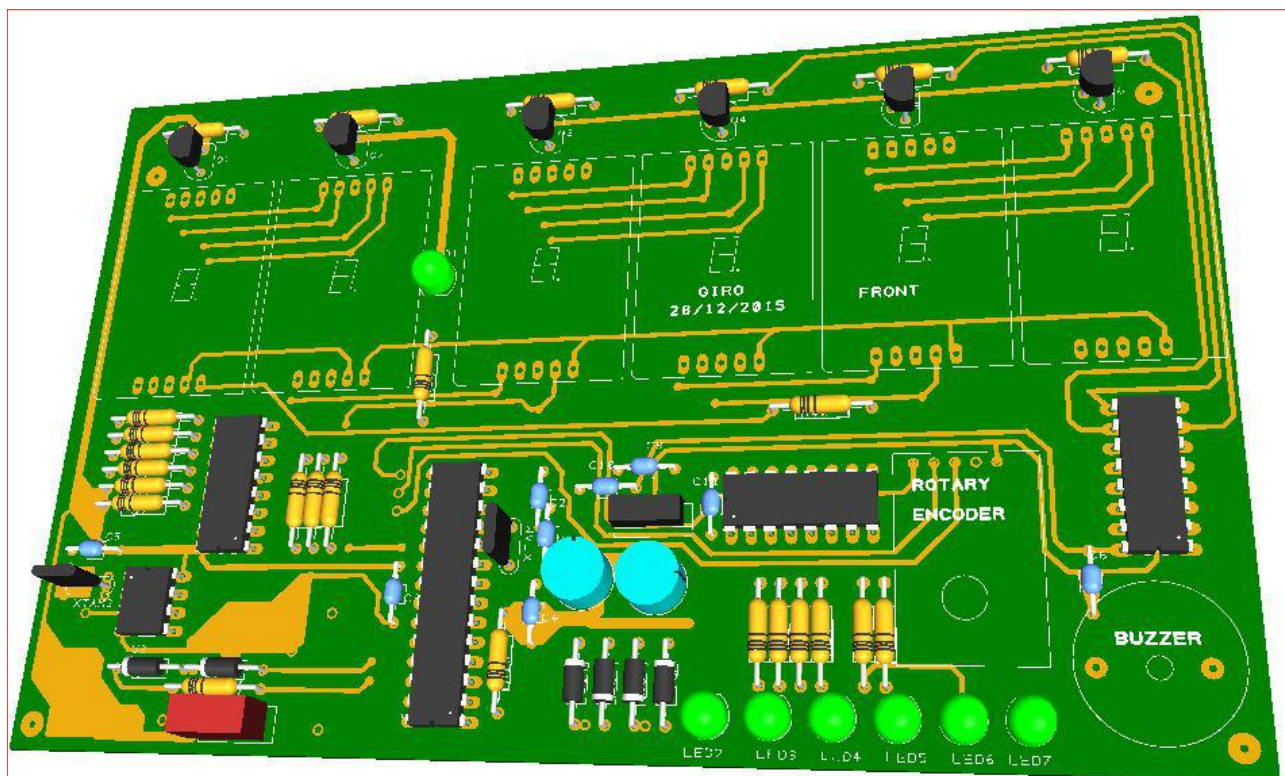
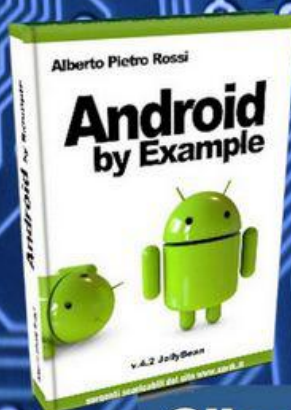
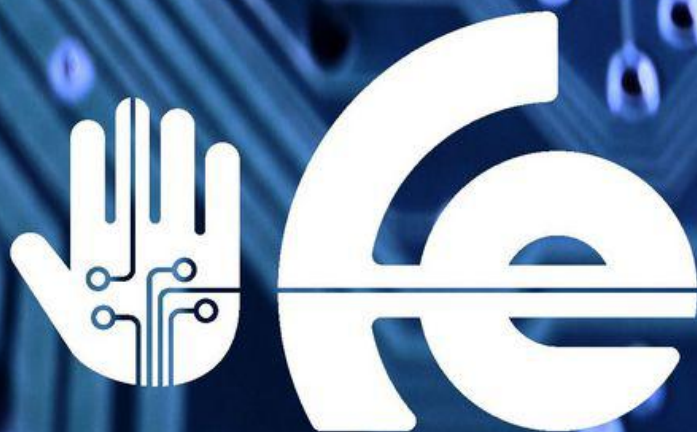



Figura 12: Realizzazione 3D V2.

START-UP: NUOVA LINFA ALL'ELETTRONICA ITALIANA

a cura della Redazione

Illuminotronica punta all'innovazione e alle nuove idee diventando vetrina di eccezione di 10 start-up tecnologiche. Illuminotronica 2016: l'unica mostra-convegno professionale italiana dedicata al visibile, in programma dal 6 all'8 ottobre a Padova – dedica ampio spazio alle nuove idee e alle soluzioni innovative. Grazie al progetto "Assodel Foundation" (creato dalla Federazione Distretti Elettronica – Italia per promuovere le eccellenze italiane), start-up e outcomer avranno la possibilità di farsi conoscere e di presentare le proprie soluzioni in fiera.

ILLUMINOTRONICA
LA FIERA SUL VISIBILE E LE SUE APPLICAZIONI
PADOVA 6-8 OTTOBRE 2016

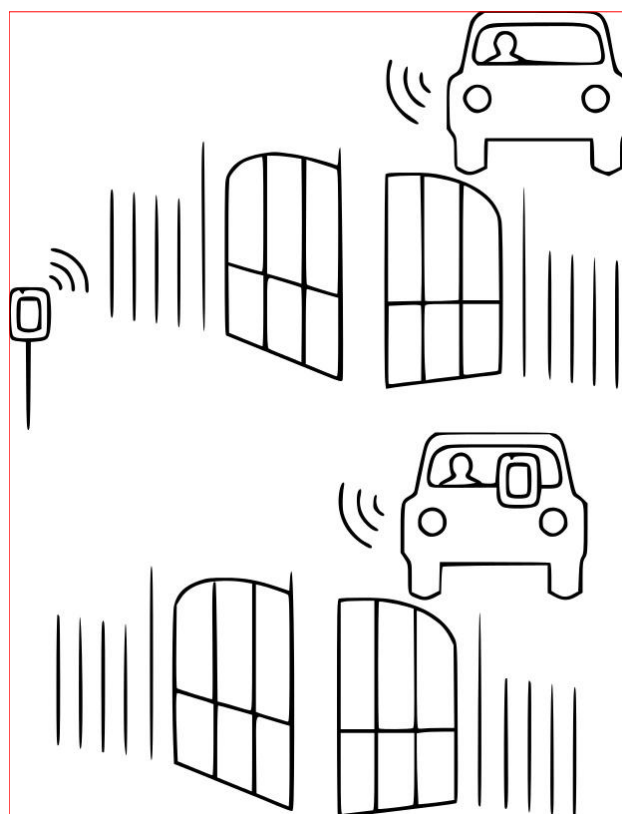
L'iniziativa

Selezionate da un comitato di esperti per qualità delle soluzioni proposte e per livello di innovazione, le start-up presentate a Illuminotronica fanno riferimento ai seguenti ambiti:

- Elettronica/IoT;
- Smart Lighting;
- Domotica;
- Sicurezza;
- Salute e benessere.

L'iniziativa è promossa da Assodel in partnership con IngDAN, i media partner dell'iniziativa sono A&V Elettronica e Fare Elettronica.

10 start-up di eccellenza

Questi i "Magnifici 10" ovvero le 10 start up selezionate mediante criteri (per citarne alcuni) di originalità dell'idea e presentazione della stessa, potenzialità di sviluppo del business, contenuto tecnologico innovativo. Saranno presenti a Illuminotronica nel Pad. 5, ciascuna con una propria postazione, e presenteranno il loro progetto durante i convegni della manifestazione. Per maggiori informazioni e per accedere al programma delle tre giornate: www.illuminotronica.it.

1Control

1. 1Control

E' una startup bresciana nata nel Novembre 2015 nell'incubatore Superpartes (www.superpartes.biz). Nonostante non abbia ancora raggiunto l'anno di età, 1Control ha già saputo ottenere importanti risultati ed è anche già presente nel parco prodotti di importanti realtà italiane della distribuzione specializzata e professionale. 1Control SOLO è un apricancello Bluetooth per smartphone, permette di aprire un cancello e garage direttamente dallo smartphone. Non serve chiamare un tecnico, non ci sono fili da collegare, è tutto molto semplice. Persino le batterie sono già inserite e garantiscono un utilizzo me-

1Control SOLO

dio di circa 2 anni. È compatibile con smartphones Apple iPhone 4S, 5, 5S, 5C, 6, 6 plus, 6S con sistema iOS 7 o superiore e Android con sistema 4.3 o superiore e Bluetooth 4.0 LE. E' compatibile con tutti i principali radiocomandi a codifica fissa e rolling-code. SOLO può essere portato in auto, ad esempio nel portaoggetti, utilizzandolo quindi in tutte le abitazioni e persino in azienda, con il solo limite di 4 automatismi per apparato. SOLO è già in vendita in tutti i principali marketplace di elettronica.

2. Limix

LiMiX Srl è una giovane e dinamica start-up italiana fondata nel marzo 2015, con sede principale a Camerino, nelle Marche. LiMiX è nata dalla forte volontà di portare la ricerca e le competenze coltivate nell'ambiente universitario in una nuova realtà aziendale per realizzare innovativi progetti industriali. LiMiX Srl è fortemente concentrata nella realizzazione di "Talking Hands", dispositivo indossabile per la traduzione della LIS, ma ha già pronti nel cassetto numerosi altri progetti e idee che spaziano dalla domotica alla sicurezza sul lavoro. Oggi in Italia vivono più di 100.000 persone non udenti, con le quali chi non padroneggia la LIS (Lingua dei Segni Italiana) difficilmente riesce a comunicare. In futuro una soluzione per ovviare questo problema potrà essere il guanto "Talking Hands", un dispositivo indossabile in grado di tradurre la LIS in voce! Talking Hands infatti registra i movimenti delle mani che riproducono il linguaggio dei segni, li traduce in tempo reale e li trasferisce ad uno smartphone, che pronuncia la frase tramite un sintetizzatore vocale. I segni prendono voce! LiMiX è certa che questo prodotto determinerà un significativo miglioramento nella qualità della vita delle persone non udenti, talvolta non pienamente integrate in alcuni ambiti sociali. Talking Hands è un prodotto di nuova concezione: al momento non esiste nulla di simile sul mercato. Il design e il software di Talking Hands hanno forti elementi innovativi che rendono il prodotto appetibile e commercialmente valido. Grande cura e attenzione è stata data allo sviluppo del design, pensato per dare

Talking Hands

Linfa

all'utente la massima libertà di movimento e indossabilità: la struttura di "Talking Hands" lascia liberi il palmo della mano e i polpastrelli, salvaguardando il tatto ed evitando problemi di riscaldamento e sudorazione della mano.

3. Robonica

Nasce dall'idea di rendere le città più sostenibili, è una start up innovativa scritta al registro delle imprese dal novembre 2014. Attualmente Robonica produce una serie di prodotti tra cui Linfa, una serra da casa totalmente automatizzata e connessa in rete, cartucce consumabili contenenti i semi e i sali per la crescita delle piante oltre a dati di coltivazione ovvero profili di crescita scaricabili dal cloud.

4. Aeromechs

E' una start up innovativa avente il suo core business nel settore della domotica, con il suo prodotto proprietario X-DOM, una soluzione semplice ed a basso costo per la domotica. X-DOM® nasce per offrire a chiunque la possibilità di rendere "un po' più domotica" la propria casa, senza grossi sforzi né installativi né economici. Il sistema, composto da moduli elettronici facilmente installabili nelle cassette 503 o quadri/sottoquadri dell'impianto elettrico, e da una app di gestione disponibile per smartphone/tablet, permette di controllare luci, serrande e tapparelle elettriche, termostati, cancelli elettronici e qualunque altro sistema provvisto di contatti per accensione e spegnimento. La comunicazione avviene tramite wi-fi, utilizzando un sistema Cloud per la trasmissione delle informazioni ed il mantenimento dello stato del sistema. Senza bisogno di centraline di controllo, o di sostanziali modifiche all'impianto elettrico di casa. Il controllo avviene sul singolo oggetto, a differenza della maggior parte delle soluzioni in commercio.

5. Apuana Corporate

E' una rete di imprese organizzate secondo un modello di business che ha già raccolto una serie di importanti riconoscimenti per l'innovazione. Principalmente la rete si occupa di trasformazione del marmo, ma privilegia la contaminazione tra le lavorazioni di tipo tradizionale e le nuove tecnologie, funzionando come piattaforma di incontro tra progettisti, produttori e mercato. Il frutto di questo lavoro trova spazio su otto marketplace fra Euro-

X-DOM

Apuana Corporate

pa e Stati Uniti rivolti al consumer con prodotti di design in marmo e alle imprese con soluzioni di illuminotecnica che prevedono l'uso del marmo di Carrara in abbinamento ai corpi illuminanti. Inoltre propone Marble Leap, un tavolo intelligente che si connette al cloud.

6. STE Industries

Nasce a marzo del 2016 e si propone sul mercato con soluzioni estremamente innovative orientate al settore emergente dell'Internet Of Things. In particolare ha brevettato una tecnologia di trasmissione dati a bassissimo consumo energetico, chiamata Micro.Sp, che consente di trasferire dati da un punto "A" (tipicamente il sensore) ad un punto "B" (tipicamente il gateway/ricevitore) con consumi decisamente più bassi rispetto ai prodotti presenti sul mercato. Per questo motivo la Vision di Ste Industries è quella di essere degli abilitatori di dati, di permetterne quindi la raccolta e la gestione attraverso piattaforme IoT di vario genere. La tecnologia Micro.Sp è coperta oggi da 5 brevetti internazionali ed è utilizzata da grandi player che propongono prodotti finiti sul mercato come ad esempio Bridgestone, Octo Telematics, Flextronics, Telenor e molti altri. La rete di sistemi wireless utilizzata è 01Net, ed è composta da sensori a basso consumo energetico e sensori a lungo raggio in grado di monitorare Co2, Temperatura, umidità, inclinazione, deformazione, sensori parking control, luminosità, etc. È prevista anche una piattaforma Cloud IoT di controllo e gestione dati in real-time.

7. BlueUp

Sistemi e tecnologie Internet of Things basate su Beacon in tecnologia BLE (Bluetooth Low Energy). BlueUp e una startup (iscritta al Registro delle Startup Innovative) fondata nel Gennaio 2014, è attiva nell'ambito delle tecnologie per l'Internet of Things, ed in particolare dei sistemi e soluzioni basati su beacon in tecnologia Bluetooth Low Energy (BLE), per applicazioni di localizzazione, identificazione e sensoristica. Un beacon BLE è un dispositivo wireless che trasmette periodicamente un pacchetto di advertising Bluetooth Low Energy, che può essere ricevuto dagli smartphone presenti in vicinanza e usato per stimare la posizione rispetto al beacon stesso. Questo consente di fornire all'utente informazioni contestualizzate, aprendo la possibilità di connettere il mondo online (virtuale) con il mondo fisico offline (reale). I beacon BlueUp sono certi-

STE Industries

Eddystone

ficati per supportare le due principali tecnologie sul mercato beacons: iBeacon, lanciata da Apple nel 2013, e Eddystone, sviluppata da Google nel 2015. BlueUp può vantare un significativo portfolio di clienti e partner che include grandi gruppi nazionali e multinazionali (Seven-Invicta, Enel Green Power, HPE, Ingenico, Gilbarco, ManutenCoop) e un gran numero di software house, digital agencies e system integrator a livello italiano ed europeo.

8. Domotica Undici

Sviluppa, produce e commercializza un sistema intelligente di controllo e gestione per l'automazione domestica e di edifici. Nata nel 2013 in seguito alla concretizzazione di un progetto iniziato nel 2011, Domotica Undici inizia le proprie installazioni subito dopo la sua costituzione. Ad oggi, con oltre 60 installazioni attive, l'azienda è in un continuo e costante trend di crescita. Progettato per l'impiego sia in ambiti residenziali che nel terziario, Undici è una soluzione wireless altamente affidabile che senza opere murarie è in grado di connettersi a dispositivi di qualsiasi natura. Il modulo end-point consente di gestire 4 ingressi e 6 uscite digitali, 2 ingressi analogici ed una uscita 0-10V. Tutto in uno. In

Undici

questo modo la progettazione avviene per numero di utenze più che per il loro tipo, abbattendo costi, ottimizzando spazi e risorse. I moduli end-point, esattamente come qualsiasi altro dispositivo che parli uno dei numerosi protocolli integrati, parlano poi con la centralina che coordina e sovraintende l'intero sistema poiché contiene tutte le logiche e le interfacce per l'utente e l'installatore.

9. Robovision Engineering

Propone ipDoor, il primo Sistema di Videocitofonia che nasce sin dalla fase di progettazione secondo lo standard IP (Internet Protocol), non solo negli apparati interni, ma anche nel Posto Esterno che ne rappresenta il cuore pulsante. Il vantaggio di questa scelta è la compatibilità intrinseca con tutto il mondo collegato ad Internet e con i suoi standard, dalle telecamere IP agli Smartphone, fino al vasto pianeta della Domotica. Altro grande vantaggio è il collegamento costante con tutti gli apparati, siano essi collegati alla rete locale, o si trovino per strada, in ufficio o dall'altra parte del mondo. In altre parole la rete è ovunque, e con ipDoor è possibile essere sempre connessi alla nostra casa o attività. Una delle maggiori innovazioni di questo sistema è la visua-

ipDoor

lizzazione dei nominativi non più su carta ma mediante targhette nominative LCD. Questa funzionalità, protetta da brevetto, consente all'utente finale di poter aggiornare i nominativi comodamente da casa o dal suo Smartphone, senza dover aprire il Posto Esterno. Sullo stesso display è possibile specificare l'indirizzo o eventualmente indicare gli orari di esercizio. Le indicazioni visualizzate sul display hanno un gradevole effetto, anche di sera, grazie alla loro luminosità automaticamente gestita dai sensori di luce e prossimità. È anche possibile l'Inserimento di un "Post-it", che sarà visualizzato accanto al nominativo o anche automaticamente pronunciato, così da comunicare al visitatore un messaggio quando effettuata la chiamata a quella persona (ad esempio, per il corriere: "rivolgersi al negozio a fianco").

10. BOL

(Beat Of Light) è una start up innovativa dedicata alla illuminazione intelligente grazie a sensori in grado di rilevare la qualità dell'aria e benessere emotivo delle persone. Il primo prodotto è una lampada di design che, collegata allo Smartphone è in grado di rilevare gli eventi che alterano il nostro equilibrio emotivo cambiando colore ed intensità di luce a seconda del livello di stress o di ansia rilevato, con il fine di andare ad agire sul benessere dell'individuo. È inoltre in grado di rilevare la qualità dell'aria (polvere sottile, temperature, umidità) nell'ambiente domestico e aziendale.

Per saperne di più

1. 1Control
http://www.1control.it/
Video: https://youtu.be/Q3M6RcwE_tI

2. Limix
http://www.limix.it/
Video: https://youtu.be/ZEZNzJNBBOU

3. Robonica
http://robonica.it/

4. Aeromechs
http://www.xdomsolutions.com/
Video: https://youtu.be/SdGAex_SmuM

5. Apuana Corporate
http://www.apuanacorporate.com/

6. STE Industries
http://www.ste-industries.com/

7. BlueUp
http://www.ipdoor.com/

8. Domotica Undici
http://www.domoticaundici.it/

9. Robovision Engineering
http://www.ipdoor.com/

10. BOL
http://beatoflight.com/